

DAA/LANGLEY

(NASA-CR-180625) NOVEL PARALLEL  
ARCHITECTURES AND ALGORITHMS FOR LINEAR  
EIGENVALUE PROCESSORS Progress Report  
(Carnegie-Mellon Univ.) 160 p Avail: NTIS  
EC AC8/MF AC1 CSCL 09B G3/61 N87-26515  
Unclas 0064467

*Langley*

1N-61

64467

*CR*

P.160

✓  
Progress Report  
Grant NAG-1-575

# NOVEL PARALLEL ARCHITECTURES AND ALGORITHMS FOR LINEAR ALGEBRA PROCESSORS

*Submitted to:*

NASA Langley Research Center  
Hampton, VA. 23665  
*ATTENTION:* John Shoosmith

*Submitted by:*

Prof. David Casasent  
Carnegie Mellon University  
Department of Electrical and Computer Engineering  
Pittsburgh, PA 15213  
*Principal Investigator:* Professor David Casasent  
Telephone: (412) 268-2464

*CR-1005*

April 1987

# Table of Contents

<b>ABSTRACT</b>	<b>1</b>
<b>1. INTRODUCTION</b>	<b>2</b>
1.1 Overview	2
1.2 Number Representation	2
1.3 Laboratory System Design, Fabrication, and Algorithms	3
1.4 Case Studies	4
1.5 Numerical Extensions	4
<b>2. BIPOLEAR BIASING IN HIGH ACCURACY OPTICAL LINEAR ALGEBRA PROCESSORS</b>	<b>5</b>
2.1 Introduction	5
2.2 The method of biasing	6
2.3 Summary	9
<b>3. OPTICAL LINEAR ALGEBRA PROCESSOR: LABORATORY SYSTEM PERFORMANCE FOR OPTIMAL CONTROL APPLICATIONS</b>	<b>10</b>
3.1 Introduction	10
3.2 Space Integrating Optical Linear Algebra Processor	10
3.3 Number Representation and Electronic Support	11
3.4 Case Study and Algorithm	13
3.5 Laboratory Results	16
3.6 Summary and Conclusion	20
<b>4. REAL-TIME OPTICAL LABORATORY LINEAR ALGEBRA SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS</b>	<b>23</b>
4.1 Introduction	23
4.2 Olap Architecture and Fabrication	23
4.3 System Properties and Use	25
4.4 Problem Definition	26
4.4.1 Explicit 1-D M-V Solution	27
4.4.2 Implicit LAE Solution	28
4.4.3 Explicit Matrix-Vector 2-D Solution	29
4.5 Case Study	30
4.6 Optical Realization Issues	32
4.6.1 Node Numbering	32
4.6.2 Partitioning and Data Flow	32
4.6.3 Partitioning	34
4.6.4 High-Accuracy Encoding	35
4.6.5 High-Accuracy Data Flow and Partitioning	36
4.6.6 Performance Measures	36
4.7 Laboratory Test Results	37
4.7.1 Implicit vs. Explicit Solutions with Computational/System Errors Included	37
4.7.2 Implicit Algorithm with Variable Time Step Size	38
4.7.3 Analog System Laboratory Performance	39
4.7.4 Encoded High-Accuracy Laboratory System Performance	39
4.7.5 Quantitative Individual High-Accuracy Multiplication Data	42
4.7.6 Graphical 2-D Temporal Temperature Data Results	43

<b>5. TIME AND SPACE INTEGRATING OPTICAL LABORATORY</b>	<b>48</b>
<b>5.1 Introduction</b>	<b>48</b>
<b>5.2 Architecture Review</b>	<b>49</b>
<b>5.3 Number Representation</b>	<b>50</b>
<b>5.4 Partitioning, LU Decomposition and Accuracy Tradeoffs</b>	<b>52</b>
<b>5.4.1 Diagonal Partitioning</b>	<b>52</b>
<b>5.4.2 Output P<sub>3</sub> Flexible Detector System</b>	<b>53</b>
<b>5.4.3 LU One-Channel Algorithm<sup>6</sup></b>	<b>54</b>
<b>5.4.4 Accuracy Above the Number of Channels by Partitioning</b>	<b>55</b>
<b>5.5 General Laboratory Electronic Support System Requirements</b>	<b>56</b>
<b>5.6 Electro-Optical Laboratory System</b>	<b>57</b>
<b>5.7 Finite Element Case Study</b>	<b>57</b>
<b>5.8 Laboratory System Data</b>	<b>60</b>
<b>5.9 Summary and Conclusion</b>	<b>63</b>
<b>6. Multi-channel Encoded System Design and Fabrication</b>	<b>65</b>
<b>6.1 Architecture</b>	<b>65</b>
<b>6.2 Electronic Support Requirements for Multi-Channel Encoded System</b>	<b>67</b>
<b>6.2.1 AO Cells and A/D Converters</b>	<b>67</b>
<b>6.2.2 Input Data Requirements</b>	<b>69</b>
<b>6.2.3 Detector P<sub>3</sub> Requirements (future and now)</b>	<b>69</b>
<b>6.3 Host Computer System</b>	<b>73</b>
<b>6.3.1 Computer System</b>	<b>74</b>
<b>6.3.2 High Speed Memories</b>	<b>77</b>
<b>6.4 Multi-Channel Encoded Processor Hardware</b>	<b>79</b>
<b>6.4.1 Clock Board</b>	<b>80</b>
<b>6.4.2 Mux/DeMux Board</b>	<b>81</b>
<b>6.4.3 ECL Shift/Add Board</b>	<b>82</b>
<b>6.4.4 100 MHz 6-bit A/D Boards</b>	<b>85</b>
<b>6.4.5 A/D Reference Supply</b>	<b>87</b>
<b>6.4.6 Four-Bit 200 MHz D/A Converter Boards</b>	<b>88</b>
<b>6.4.7 RF Drivers and Oscillators</b>	<b>89</b>
<b>6.5 Detector Array and Fiber Optic Coupling</b>	<b>91</b>
<b>6.6 Software</b>	<b>93</b>
<b>6.7 Low Level Routines</b>	<b>95</b>
<b>6.7.1 Data Handling Conventions</b>	<b>95</b>
<b>6.7.2 Hardware Dependent Routines</b>	<b>98</b>
<b>6.7.3 Scalar Multiply Routines</b>	<b>99</b>
<b>6.7.4 LU Decomposition Software</b>	<b>100</b>
<b>6.7.5 Software list</b>	<b>100</b>
<b>6.8 Initial Laboratory System</b>	<b>101</b>
<b>6.9 Single Bit Test System</b>	<b>101</b>
<b>6.9.1 Construction</b>	<b>101</b>
<b>6.9.2 Results</b>	<b>107</b>
<b>6.10 Three Channel System</b>	<b>108</b>
<b>6.10.1 Construction</b>	<b>108</b>

6.10.2 Results	111
6.11 Summary	114
<b>7. LABORATORY OLAP PERFORMANCE AND PLANS</b>	<b>118</b>
7.1 Laboratory OLAP Characterization	118
7.1.1 Laboratory OLAP System Review	118
7.1.2 System Performance Limitations	121
7.2 AC-Coupled OLAP	123
7.2.1 AC Coupling Basics	124
7.2.2 Laser Diode Modulation	124
7.2.3 Laser Diode Imaging Optics	126
7.2.4 AC Coupled Detector System	127
7.3 Future Plans	128
<b>8. CASE STUDIES FOR SIMULATION AND TESTNG OF THE OPTICAL LINEAR ALGEBRA PROCESSOR</b>	<b>129</b>
8.1 Introduction	129
8.2 Computational Fluid Dynamics	129
8.2.1 Nonlinear, Steady-State CFD	130
8.2.2 Nonlinear, Transient CFD	134
8.3 CFD Summary	135
8.4 Linear Dynamic Structural Mechanics Case Study	136
8.4.1 Linear Dynamic Analysis	138
<b>9. OPTICAL PROCESSING EXTENSIONS</b>	<b>140</b>
<b>10. SUMMARY AND FUTURE WORK</b>	<b>142</b>
<b>I. POST-DETECTION HARDWARE DESIGN</b>	<b>143</b>
I.1 Introduction	143
I.2 Basic OLAP output hardware	143
I.2.1 Case 1: radix is positive or negative and a power of 2	145
I.2.2 Case 2: Radix is positive or negative and not a power of 2	147
I.3 Future OLAP Detection Systems	150
I.4 Conclusions and Summary	152

## List of Figures

<b>Figure 3-1:</b>	Simplified schematic of the space integrating optical linear algebra processor.	14
<b>Figure 3-2:</b>	Details of one point modulator laser diode and collimating optics system.	14
<b>Figure 3-3:</b>	Photograph of the optical laboratory system.	15
<b>Figure 3-4:</b>	Diagram of the electronic support system	17
<b>Figure 3-5:</b>	Photograph of the electronic support system.	18
<b>Figure 3-6:</b>	Data flow arrangement for partitioning	19
<b>Figure 3-7:</b>	Linear analog performance of 10 bits (0.1%).	20
<b>Figure 3-8:</b>	Linear analog frequency-multiplexed performance.	21
<b>Figure 3-9:</b>	Frequency-multiplexed optical interconnection architecture.	22
<b>Figure 4-1:</b>	Simplified schematic of the space integrating optical linear algebra processor.	25
<b>Figure 4-2:</b>	Structure for the matrix in the implicit 2-D diffusion equation solution	31
<b>Figure 4-3:</b>	Multi-processor architecture for multiple banded matrices	33
<b>Figure 4-4:</b>	Data sequence for updating of explicit M-V formulation of 2-D diffusion PDE	34
<b>Figure 4-5:</b>	Data flow for the high-accuracy multiplication (the example shown is for the product of two 5-digit numbers)	35
<b>Figure 5-1:</b>	Multi-Channel High Accuracy Time and Space Integrating Architecture <sup>6</sup>	50
<b>Figure 5-2:</b>	One-Channel LU Decomposition Architecture for Matrix Decomposition <sup>6</sup>	52
<b>Figure 5-3:</b>	Output $P_3$ Detector System with Number Representation and Component Flexibility and with a New Conversion Algorithm Ability	53
<b>Figure 6-1:</b>	Multi-Channel AO Cell Architecture	66
<b>Figure 6-2:</b>	Diagram of acoustic gaps from $P_1$ and data inputs to $P_2$	68
<b>Figure 6-3:</b>	Diagram of the Mixed-Binary to Binary Converter	72
<b>Figure 6-4:</b>	Operation Performed by Mixed-Binary to Binary Converter	73
<b>Figure 6-5:</b>	System Block Diagram	75
<b>Figure 6-6:</b>	Photograph of the Intel 86/380 System	76
<b>Figure 6-7:</b>	Board Layout for Intel Multibus 1 Chassis	77
<b>Figure 6-8:</b>	High Speed Analog Hardware, Block Diagram	80
<b>Figure 6-9:</b>	Picture of the ECL shift/add board	83
<b>Figure 6-10:</b>	Control Section Block Diagram	84
<b>Figure 6-11:</b>	Photograph of one 6 Bit, 100 MHz A/D board	87
<b>Figure 6-12:</b>	Four bit D/A generating a random output	90
<b>Figure 6-13:</b>	Photograph of the detector box used	94
<b>Figure 6-14:</b>	Photograph of the SELFOC/fiber-optic setup	94
<b>Figure 6-15:</b>	9 Bit Multiply on an 3 Bit Optical System	96
<b>Figure 6-16:</b>	Assumed Decimal Point Handling	98
<b>Figure 6-17:</b>	Single Channel Test System	102
<b>Figure 6-18:</b>	Detector output from the timing setup program	106
<b>Figure 6-19:</b>	Test outputs from the single channel test system	108
<b>Figure 6-20:</b>	Example RF inputs to the AO cells	113
<b>Figure 6-21:</b>	Example detector outputs	113

<b>Figure 6-22:</b>	Action performed by the shift/add on the example problem	114
<b>Figure 6-23:</b>	Example mixed binary outputs	115
<b>Figure 6-24:</b>	Example system final output	115
<b>Figure 7-1:</b>	Laboratory Optical System Schematic	119
<b>Figure 7-2:</b>	Laboratory System Block Diagram	120
<b>Figure 7-3:</b>	Laser Diode Operation Curve	125
<b>Figure 8-1:</b>	Flow in a Driven Cavity	130
<b>Figure 8-2:</b>	Discretized Driven Cavity Domain	131
<b>Figure 8-3:</b>	Beam element	136
<b>Figure 8-4:</b>	Case study structure model	137
<b>Figure I-1:</b>	OLAP output configuration	144
<b>Figure I-2:</b>	Case 1: Forming binary $Z$ from $N$ binary words, $c_i$	146
<b>Figure I-3:</b>	Shift/Add procedure when $R=3$	148
<b>Figure I-4:</b>	Expressing $Z$ of base $r=2,3$ , etc. as base $r=+2$ .	149
<b>Figure I-5:</b>	OLAP detection system and conversion unit	150
<b>Figure I-6:</b>	Future back-end hardware	151
<b>Figure I-7:</b>	Future post-detection hardware with two-levels of CCDs	152

## ABSTRACT

Research done at the Carnegie Mellon Center for Excellence in Optical Data Processing for Nasa Langley is reviewed, and the work proposed for the third year is detailed. The report covers number representations, processing architectures and algorithms, optical linear algebra processor fabrication and test results, case study descriptions, and future system plans.

# 1. INTRODUCTION

## 1.1 Overview

This report describes the current status of the research work done at the Carnegie Mellon Center for Excellence in Optical Data Processing for Nasa Langley. This chapter is an overview of the report. In the first year of the research contract, progress was made in developing: new number representations, a new processing architecture and LU decomposition algorithm, and error source modelling. In the second year, fabrication and test of a prototype system was performed, along with extensions from some of the first year topics. These research efforts are detailed herein. Much was learned in the development and operation of the prototype system, and an evaluation of the system was made resulting in an improved laboratory processing system. New numerical extensions of the optical system are proposed.

Brief summaries of the topics of this report follow in the rest of this chapter. A detailed explanation is provided in subsequent chapters.

## 1.2 Number Representation

The processing of bipolar data is an important issue for optical data processing systems. Most optical processing architectures modulate the intensity of light. Since this intensity cannot be "negative", bipolar data must somehow be incorporated into the processing. In the first year of research, two methods of bipolar data encoding were developed. The first method was based on twos-complement encoding<sup>1</sup>, with new processing in the back end of the processor included to improve the efficiency of conventional twos-complement encoding techniques<sup>2,3</sup>. The second method used negative base encoding for processing bipolar data<sup>4</sup>. In this report we describe a third method which is based on biasing the input data to the processor. This method is detailed in Chapter 2.

### 1.3 Laboratory System Design, Fabrication, and Algorithms

The step-by-step development of the laboratory systems and the use of various algorithms and case studies is an important aspect of our optical computing research. It helps to better evaluate our system design, error source models, and many practical issues. The first acousto-optic (AO) based processor fabricated<sup>5</sup> was a five-channel analog frequency-multiplexed processor. This processor was used to obtain an iterative analog solution to a matrix-vector problem. This is described in Chapter 3. The processor was then used to implement an explicit solution method to solve a parabolic PDE case study, as described in Chapter 4.

We now focused our attention on case studies which require implicit solution methods, i.e. those which often yield the more stable and accurate results. We also moved to encoded number representations on the laboratory system. This provided us with a reduced dynamic range requirement for the processor, and thus much more tolerance of processor error sources. We concentrated on a new multi-channel system architecture<sup>6</sup>, and fabricated a small cross-section of the full multi-channel system, which was our prototype processor. We demonstrated this new laboratory system by running a structural dynamics finite element plate bending case study on the processor. The description of the laboratory system, data, performance, and other details is provided in Chapter 5. The fabrication of the prototype processor, including optics and electronics, and the software control of the system are described in Chapter 6. Use of the laboratory system resulted in an evaluation and recommendations for a new architecture to eliminate some of the problems with the current one. This is discussed in Chapter 7.

Some of the new features described in Chapters 5, 6, and 7 include:

- Demonstrated partitioning of a large problem on a small processor.
- Successfully processed digitally encoded data.
- Used partial product partitioning to process word lengths larger than the number of hardware channels at  $P_2$ .

- Implemented a direct solution method on an optical processor.
- Laboratory demonstration of a new one channel LU decomposition algorithm.
- Handled bipolar data with a sign-magnitude representation on the one channel processor.

## 1.4 Case Studies

Three new case studies have been developed for further implementation and study on the laboratory optical processor. Two case studies are taken from computational fluid dynamics (CFD). One is a finite element formulation and the other is a finite difference problem. The third case study is a finite element problem taken from structural mechanics. The case studies are detailed in Chapter 8.

## 1.5 Numerical Extensions

Optical systems can perform other numerical functions, and we specifically describe polynomial evaluation and on-line arithmetic. This description is given in Chapter 9. Such numerical extensions involve using a general purpose back end hardware. Appendix A details the hardware realization possible for a general purpose back end for different number representations. Currently, we perform these tasks in software, using the existing back end hardware described in Chapters 5 and 6. No additional work on this is planned in the third year of our research. On-line arithmetic will be detailed in year 3, but we do not currently plan a hardware implementation of it.

## 2. BIPOLEAR BIASING IN HIGH ACCURACY OPTICAL LINEAR ALGEBRA PROCESSORS

### 2.1 Introduction

In this chapter we propose a method of biasing data as a means of handling bipolar data in high-accuracy optical linear algebra processors (OLAPs). Biasing converts matrix-vector operations from bipolar to unipolar and is shown to be more efficient than several other methods including two's complement and time-multiplexing.

Recently, much interest has been given to the use of optics as a means of performing various linear algebra operations<sup>7</sup>. Optical Linear Algebra Processors (OLAPs) have been presented in many differing architectural designs<sup>7</sup>. The high-accuracy OLAP systems treat *digital multiplication by analog convolution* (DMAC)<sup>8, 9, 10</sup> as the preferred algorithm. To date, the methods discussed for handling bipolar data in high-accuracy OLAPs include two's complement<sup>2</sup>, sign-magnitude<sup>6</sup>, space or frequency multiplexing<sup>5</sup>, and time-multiplexing<sup>5, 11</sup>. Many articles have ignored the subject of bipolar data altogether. Each of the above methods have limitations. The two's complement method requires, in general, N additional bits for an N-bit word (this is wasteful of space bandwidth product, SBWP) and requires twice the amount of electronic support to handle bipolar data. Similar remarks apply to space-multiplexing methods. Time-multiplexing methods work by processing the positive and negative data separately and thus reduce the processing speed by a factor of two. Such methods also require more complicated output storage and data combinations. Sign-magnitude approaches are not extendable to multichannel systems where vector inner products (VIPs) are formed by the addition of separate products via space integration. These multichannel processors, where each channel performs one multiplication to create a VIP term, are essential to provide sufficiently parallel systems with large enough operations-per-second speeds to be competitive with digital

systolic and other approaches. Thus, methods to handle bipolar data in such multi-channel processors are felt to be essential. In this chapter, we discuss the use of *biasing* as a means of handling bipolar data in high-accuracy multi-channel OLAP architectures. We show that this method is easily implemented and extends to non-binary bases. Use of a non-binary base has recently been shown to be suitable for optical realization and most efficient in the use of SBWP and electronic support<sup>6</sup>.

## 2.2 The method of biasing

The purpose of biasing is to convert a *bipolar* matrix-vector operation into a *unipolar* one. The advantage to such a system is obvious; all integer or floating-point values within the processor are strictly positive thus eliminating the need for sign encoding. All prior discussions of biased data have concerned analog processors. This chapter addresses multi-channel and high-accuracy OLAP systems using encoded data representations.

In the DMAC algorithm, the bits of two encoded numbers are convolved to form the product of the two numbers in mixed binary representation. The output is easily converted to conventional binary by A/D converting each output bit and adding it (shifted) to the next most-significant-bit (MSB). The bias method presented here is applied to such encoded data, is new and has many attractive properties. The algorithm creates strictly positive, "biased" data from the original OLAP input data. Any radix encoding employed is unaffected by the biasing. The choice for the bias term,  $b$ , is not arbitrary but depends on the most negative value of the original input data. In addition, the output data from the biased system is altered from that of the original output data. Thus, a correction term which we will call  $b\Delta$  must be computed and subtracted from the biased output. The result of the subtraction is the desired bipolar processor output. Briefly, negative valued data can appear prior to and after optical operations whereas manipulations on optical data within the OLAP are strictly on positive-valued data.

As an example on which we will base our discussion, let us assume the OLAP performs a matrix-vector multiplication of the form

$$\mathbf{Ax} = \mathbf{c} \quad (2.1)$$

where  $\mathbf{A}$  is an  $n \times n$  matrix and  $\mathbf{x}$  and  $\mathbf{c}$  are  $n \times 1$  column vectors. The matrix-vector elements, i.e.,  $a_{i,j}$  and  $x_j$ , are assumed to be bipolar and binary encoded. In order that the biased matrix-vector data be positive unipolar, the bias  $b$  must be a value greater than or equal to the magnitude of the *most negative* element in  $\mathbf{A}$  or  $\mathbf{x}$  ( $b$  is always a positive number). Every nonzero element of  $\mathbf{A}$  and  $\mathbf{x}$  is then incremented by  $b$ , thus creating a biased matrix  $\mathbf{A}_b$  and vector  $\mathbf{x}_b$  whose elements are  $(a_{i,j} + b)$  and  $(x_j + b)$  respectively, and which are strictly positive. Zero valued elements in  $\mathbf{A}$  and  $\mathbf{x}$  are not incremented, thus retaining any sparse or banded structure that may exist. The OLAP now performs the matrix-vector multiplication

$$\mathbf{A}_b \mathbf{x}_b = \mathbf{c}_b. \quad (2.2)$$

where the output vector  $\mathbf{c}_b$  differs from the desired vector  $\mathbf{c}$  by a term  $b\Delta$  which depends on the bias  $b$  and the elements of  $\mathbf{c}$ . The relation between the two is given by

$$\mathbf{c} = \mathbf{c}_b - b\Delta \quad (2.3)$$

where  $\Delta$  is a vector of length  $n \times 1$  and termed the *correction vector*. It can easily be shown that the elements,  $\delta_i$ , of  $\Delta$  are given by

$$\delta_i = \sum_{j=1}^n (a_{i,j} + x_j) + p_i b. \quad (2.4)$$

We envision an optical processor that computes the matrix-vector product by a sequence of VIP operations, i.e., one element of  $\mathbf{c}$  sequentially. In such a formulation,  $p_i$  is the number of nonzero product terms in each ( $i$ ) unbiased VIP ( $p_i$  is less than or equal to  $n$ , depending on the number of zeros in a given row of  $\mathbf{A}$  and the vector  $\mathbf{x}$ ). Thus each  $\delta_i$  is the sum of the elements that are multiplied to produce  $c_i$ , plus  $p_i$  times the known bias. These  $a_{i,j}$  and  $x_j$  are known *a priori* and hence each  $\delta_i$  can easily be calculated in external adder circuitry (including sign encoding since  $\delta_i$  may be negative) simultaneous with the optical formation of  $\mathbf{c}_b$ . The subtraction of  $b\delta_i$  from the computed output element  $i$  of the VIP results in the desired VIP elements  $c_i$  in (2.1).

We now show that this bias technique applies to any encoded data using the DMAC algorithm. We also show that no loss in bit accuracy is incurred. We assume that the unbiased elements of (2.1) are binary encoded in N bits and extend to both positive and negative values. By choosing the bias  $b$  to exactly equal the magnitude of the most negative element in  $\mathbf{A}$  or  $\mathbf{x}$  the range of biased data extends from zero to  $\max[a_{i,j}, x_j] + |\min[a_{i,j}, x_j]|$ , where the second term is the bias  $b$  and where  $\min[a_{i,j}, x_j]$  is negative. A larger value of  $b$  would increase the number of required bits in (2.2) and hence the optimum choice for  $b$  is  $|\min[a_{i,j}, x_j]|$ . Under worst case conditions,  $\max[a_{i,j}, x_j] = |\min[a_{i,j}, x_j]|$  and the data is symmetric about zero. The largest biased element is then  $2(\max[a_{i,j}, x_j])$ . In order that this maximum value be representable in the N bits of the OLAP, the magnitude of the unbiased data must be restricted to N-1 bits. Hence, to form the biased data representations we require one extra bit in each matrix and vector element of (2.2). However, conventional bipolar data encoding schemes require at least one additional sign bit, so that biasing suffers no relative loss in data range (in terms of the number of bits required). Since the data are encoded after biasing, we also observe that the dynamic range of the optical system is unchanged from that of the unbiased system.

We have shown that biasing creates a unipolar OLAP problem from a bipolar one. Because the biased and unbiased data are encoded in the same radix, DMAC is unaffected by the biasing technique. The DMAC algorithm, operating on biased data, produces the linear algebra result of (2.2), and its correction by  $b\Delta$  as in (2.3), results in the desired output of (2.1). This same combination of DMAC and biasing can easily be extended to any non-binary base (radix). Also, since the biased OLAP operates on only positive data, biasing is directly applicable to multi-channel systems where multiple scalar products are summed onto a single detector<sup>6</sup>.

### **2.3 Summary**

We have considered the realization of a multi-level biasing method for high-accuracy optical linear algebra processors. Its purpose is to eliminate the need for sign encoding during optical processing. Our proposed bias method does not require any additional bits relative to other bipolar encoding schemes and suffers no loss in dynamic range or in the data range that it can handle. Biasing is equally applicable for multichannel OLAPs where the output is a VIP formed by the addition of separate scalar products. In general, it may be said that the method of biasing presented in this paper represents a technique which is easily implemented and applicable to many OLAP systems where unipolar processing is required.

### 3. OPTICAL LINEAR ALGEBRA PROCESSOR: LABORATORY SYSTEM PERFORMANCE FOR OPTIMAL CONTROL APPLICATIONS

#### 3.1 Introduction

A space integrating optical linear algebra processor is described and laboratory performance of the system in the solution of nonlinear matrix equations for optimal control are presented. A new matrix partitioning method is described and the accuracy of the analog implementation of this processor is emphasized. This same architecture is capable of high accuracy performance. Different performance measures and their suitability as criteria for performance are also noted and discussed.

Many Optical Linear Algebra Processors (OLAPs) have been suggested<sup>7</sup>, but few have been fabricated. One such well-engineered system that has been fabricated is a space integrating and space multiplexed architecture whose electronic support system and initial operation was recently described<sup>5</sup>. In Section 3.2, we review the processor, its fabrication and how bipolar and complex-valued data are handled on this system. In Section 3.3, the high accuracy and analog performance of the system, partitioning, and the electronic support system are addressed. Our case study and algorithm are then advanced in Section 3.4 and laboratory results are then included in Section 3.5.

#### 3.2 Space Integrating Optical Linear Algebra Processor

The space integrating OLAP is shown schematically in Figure 3-1. At  $P_1$ , separate linear arrays of point modulators are placed. These are imaged onto an Acousto-Optic (AO) cell at  $P_2$  and the Fourier transform of the light leaving  $P_2$  is collected on detectors at  $P_3$ . Two linear arrays are shown at  $P_1$ . These are fed with the positive  $\underline{a}^+$  and negative  $\underline{a}^-$  elements of the input vector  $\underline{a}$ . The AO cell at  $P_2$  is fed with the vector  $\underline{b}$  frequency-multiplexed with its three

unipolar projections at  $0^\circ$ ,  $120^\circ$ , and  $240^\circ$  in the complex plane, thus allowing complex-valued data vector  $\underline{b}$  information to be handled. For the case when  $\underline{a}$  is complex-valued, three linear input arrays would be used. The light leaving  $P_2$  contains the products of  $\underline{a}^+$  and the three  $\underline{b}$  components  $\underline{b}^{(1)}$ ,  $\underline{b}^{(2)}$ , and  $\underline{b}^{(3)}$  traveling downward and leaving  $P_2$  at three different angles corresponding to the three multiplexed frequencies. The products of  $\underline{a}^-$  and the three  $\underline{b}$  components leave  $P_2$  traveling upward at the same three frequencies. The six point-by-point products are summed by the output integrating lens onto six separate detectors at  $P_3$ .

The system is thus a space integrating frequency-multiplexed processor with only six output detectors and with local (at the AO cell) and global (the output integrating lens) connections. Bipolar (and complex-valued) input  $\underline{a}$  vector data is represented by space-multiplexing at  $P_1$  and for  $\underline{b}$  data by frequency-multiplexing at  $P_2$ . The input point modulator system consists of individual laser diodes with separate collimating optics (Fig. 3-2). The output from these input point modulators is reduced by the imaging optics between  $P_1$  and  $P_2$  of Fig. 3-1 to match the size of the data packets in the AO cell at  $P_2$ . We denote the time separation between separate data packets by  $T_B$  (this also corresponds to the time interval at which data is fed to the  $P_1$  point modulators). To accommodate the spacing of the output detectors, a faceplate with Selfoc lenses coupled by fibers to the detectors is employed. A photograph of the optical laboratory system is shown in Fig. 3-3. It presently occupies approximately 3 feet by 2 feet on an optical bench, however this size can clearly be reduced.

### **3.3 Number Representation and Electronic Support**

This architecture is unique since it can operate analog or to high accuracy. In the analog mode, the system is linear to 9 bits. This is achieved by correcting for all static errors in the system. All such errors are correctable as we have noted in earlier publications. The linear analog performance of the system is presently limited by detector noise, electronic temporal

coupling, and temporal drift. To operate this system to high accuracy, the data inputs are encoded and the  $P_1$  inputs are fixed while the  $P_2$  data moves through the AO cell. The  $P_3$  outputs are thus the convolution of the two data bit streams and hence high accuracy performance is obtained by the Digital Multiplication by Analog Convolution (DMAC) algorithm<sup>9, 10</sup>. To achieve best performance, we operate DMAC with N digits and L levels and thus achieve  $L^N$  accuracy. With  $N = 10$  and  $L = 7$ , we achieve 28 bit accuracy with only 10 digits or  $P_1$  point modulators. With 10 modulators per row at  $P_1$  and input data at 10 Mhz per channel, this system performs 20 multiplications and additions per 0.1  $\mu$ s or 200 MOPs (complex multiplications and additions).

The electronic support system is quite general purpose and well engineered. A 68,000 control processor running UNIX is used for support. It contains 512K bytes of no-wait memory and 512K bytes of multibus memory. The system also contains its own 20M byte disk and an 0.5" 1600 BPI tape drive. This support system and processor is thus quite self-contained. It can be down-loaded with data from a VAX. The input data to  $P_1$  and  $P_2$  is buffered in the high-speed parallel memory (12 bits per channel, 8 channels per board, 10 MHz per word per channel). Separate high speed 12 bit 10 MHz D/As are present on each memory output channel and  $P_1$  and  $P_2$  input. The system's  $P_3$  output data is similarly processed with parallel A/D (12 bit, 20 MHz) and memory boards for each detector output. The general diagram of the digital support facility is shown in Fig. 3-4. It also includes video terminals, video boards and a display. A photograph of the electronic support system is shown in Fig. 3-5.

We operate the system to maintain optimum data flow. Another attractive aspect of this system is its ability to handle matrix and vector problems larger than the size (the number of point modulators at  $P_1$ ) of the system. One can achieve partitioning of such large problems by feeding the matrix elements diagonally to  $P_1$  and partitioning the problem diagonally, with

subsequent diagonal data flow. In Fig. 3-6, we show an alternative and preferable partitioning scheme. We consider the multiplication of a nine element vector by a  $9 \times 9$  matrix on a system with five input point modulators  $P_1$ . The vector data is fed to the AO cell at  $P_2$  and repeated twice. The five input point modulators at  $P_1$  are fed with five different matrix elements at successive time intervals  $nT_B$ . In Fig 3-6, the matrix elements are labeled with numbers from 0 to 18 denoting the order at which different groups of matrix elements are fed to the  $P_1$  laser diodes. The numbers associated with each group of matrix elements correspond to the time intervals  $1T_B$  to  $18T_B$ . The associated system outputs are combined as noted beside the table.

### 3.4 Case Study and Algorithm

The case study chosen was an optimal control problem, i.e. the calculation of the optimal controls to minimize a quadratic performance index for a linear quadratic regulator problem. This involves solution of a nonlinear matrix equation, the algebraic Riccati equation, for  $S$ ,

$$\underline{S} \underline{F} + \underline{F}^T \underline{S} - \underline{S} \underline{G} \underline{R}^{-1} \underline{G}^T \underline{S} + \underline{Q} = \underline{0}. \quad (3.1)$$

We solve this using the Kleinman algorithm

$$\underline{S}(k) \underline{F}(k) + \underline{F}^T(k) \underline{S}(k) = -[\underline{S}(k-1) \underline{G} \underline{R}^{-1} \underline{G}^T \underline{S}(k-1) + \underline{Q}]. \quad (3.2)$$

To solve (3.2) for  $\underline{S}$ , we convert (3.2) to a system of linear algebraic equations by lexicographically ordering the matrix  $\underline{S}(k)$  as the vector  $\underline{x}(k)$ . The solution for  $\underline{x}$  thus requires solution of the linear algebraic equation

$$\underline{H}(k) \underline{x}(k) = \underline{y}(k), \quad (3.3)$$

for  $\underline{x}(k)$ . This is first done for  $k = 1$ . Then  $\underline{x}(1)$  is used to update and calculate the new  $\underline{H}(k+1)$ . The new Eq. (3.3) with  $k = 2$  is then solved for  $\underline{x}(2)$ . And the process is repeated. We denote the outer (Kleinman) loop step index by  $k$ . Using a recursive Richardson solution to (3.3) for each iteration  $k$  (with  $r$  being the Richardson index), we solve (3.3) for each  $k$  using

$$\underline{x}(r+1) = (\underline{I} - \omega \underline{H}) \underline{x}(r) + \omega \underline{y}. \quad (3.4)$$

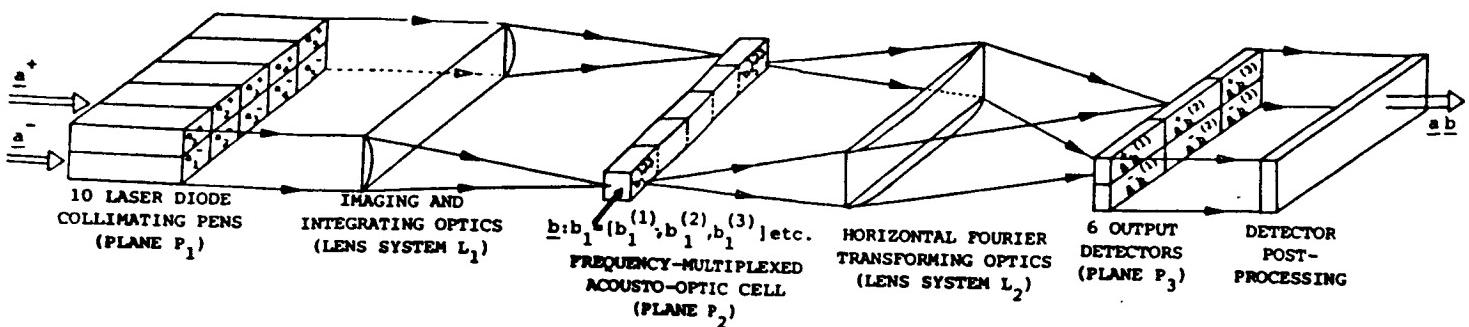


Figure 3-1: Simplified schematic of the space integrating optical linear algebra processor.

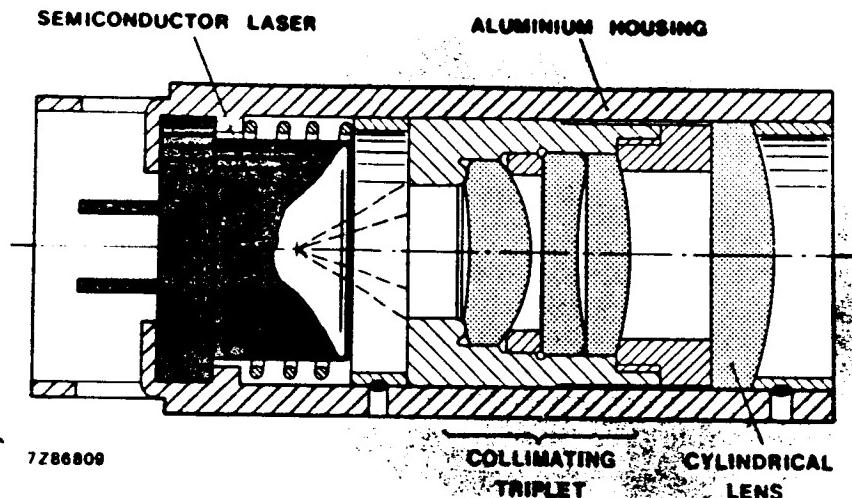


Figure 3-2: Details of one point modulator laser diode and collimating optics system.

The specific problem chosen concerned a F100 turbofan jet engine, an  $N \times N$  matrix  $\underline{H}$ , and an  $N \times 1$  vector  $\underline{y}$ , with  $N = 9$ . The matrix  $\underline{H} (5)$  after the fifth Kleinman loop is

$$\begin{bmatrix} -0.657 & -0.083 & 0.404 & -0.083 & 0.000 & 0.000 & 0.404 & 0.000 & 0.000 \\ 0.063 & -0.470 & -0.021 & 0.000 & -0.083 & 0.000 & 0.000 & 0.404 & 0.000 \\ -0.138 & -0.087 & -0.828 & 0.000 & 0.000 & -0.083 & 0.000 & 0.000 & 0.000 \\ 0.063 & 0.000 & 0.000 & -0.470 & -0.083 & 0.404 & -0.021 & 0.000 & 0.404 \\ 0.000 & 0.063 & 0.000 & 0.063 & -0.284 & -0.021 & 0.000 & -0.021 & 0.000 \\ 0.000 & 0.000 & 0.063 & -0.138 & -0.087 & -0.642 & 0.000 & 0.000 & -0.021 \\ -0.138 & 0.000 & 0.000 & -0.087 & 0.000 & 0.000 & -0.828 & -0.083 & 0.404 \\ 0.000 & -0.138 & 0.000 & 0.000 & -0.087 & 0.000 & 0.063 & -0.642 & -0.021 \\ 0.000 & 0.000 & -0.138 & 0.000 & 0.000 & -0.087 & -0.138 & -0.087 & -1.000 \end{bmatrix} \quad (3.5)$$

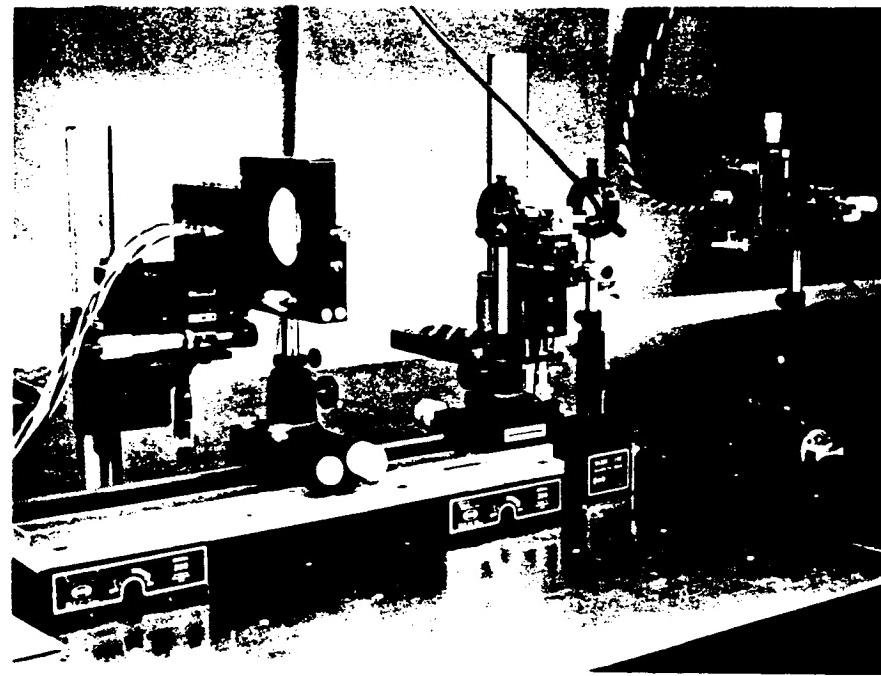


Figure 3-3: Photograph of the optical laboratory system.

and the state vector  $\underline{y}(5)$  is

$$\begin{bmatrix} -0.05626 \\ -0.01295 \\ -0.05076 \\ -0.01295 \\ -0.00304 \\ -0.01164 \\ -0.05076 \\ -0.01164 \\ -0.16297 \end{bmatrix} \quad (3.6)$$

The acceleration parameter used was determined from the Euclidean norm as  $\omega = -1.207$  to ensure that all eigenvectors of  $I - \omega H$  lie within the unit circle.

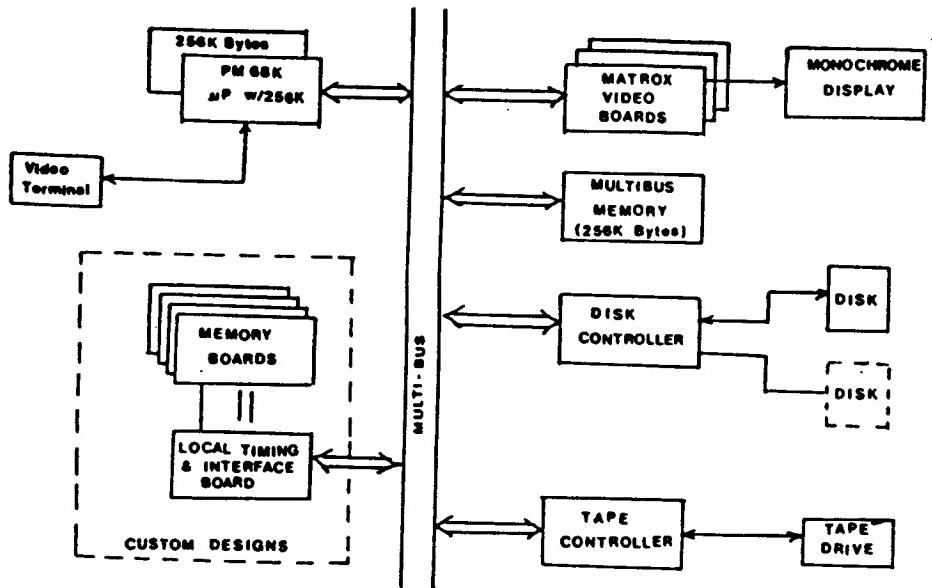
### 3.5 Laboratory Results

Figure 3-7 shows the linear analog performance of the system. The three laser diode (LD) inputs were ramps in time over the 4096 level range allowed by our D/A converters (top figure). The RF input to the AO cell contained three regions (opposite the three LD inputs respectively) at 1/6, 1/3 and full power (central figure). The three detector outputs (lower figure) show the products of the LD ramp input and the three different RF levels. The accuracy measured was 10 bits (0.1%). Due to temperature drift and temporal effects, on-line system performance is typically nine bits. Figure 3-8 shows the linearity and frequency-multiplexing performance of the analog system. The laser diode inputs were ramps (top figure). Two multiplexed frequencies to the AO cell were used and fed with the uniform half strength signal on frequency one (see second figure) and with a full and one-third power signal present in different regions on frequency two (see the third figure). Detector output one (see the fourth figure in Fig. 3-8 and detector two output (see the fifth figure in Fig. 3-8) are the products of a laser diode ramp and the associated RF signals. This demonstrates the accuracy of the system under frequency-multiplexing. The two frequencies used here were 175 MHz and 225 MHz. In other tests and demonstrations, the high accuracy of the system with base two and with higher radices has been demonstrated and quantified.

Table 3-1 summarizes four of the test results obtained on the system of Fig. 3-1 in the solution of (3.3) for the F100 problem described in Section 3.4. The performance measures used were the fractional error  $\Delta x$  in the solution vector and the fractional error in the closed loop poles  $\Delta \lambda$ . The  $\Delta \lambda$  measure is the preferable one, since it describes the regulated control system we consider. Our goal was to obtain a  $\Delta \lambda$  accurate within 1-2%. This is quite acceptable and compatible with the accuracy with which the parameters of such control models are selected. We include both performance measures to note that larger errors in the computed vector can be obtained with adequate  $\Delta \lambda$  error resulting. Test 1 is a time-multiplexed version of the system,

**TABLE 3-1:**  
Laboratory (Tests 1-3) and Simulated (Test 4) data results

Test No.	Code	Fractional Error $\Delta x$ in Soln. Vector	Fractional Error in Closed-Loop Poles $\Delta x$	Remarks
1	LQRL.txt	0.062	0.014	---
2	LQRMduty.txt	0.048	0.009	Reduced LD Drift
3	LQROmux.txt	0.071	0.014	Freq.-Multiplexing
4	VIPAE.txt	0.075	0.023	Simulation



**Figure 3-4:** Diagram of the electronic support system

in which bipolar data is accommodated by using a system twice, and subtracting the outputs on successive cycles. In Test 2, the system was operated at a reduced duty cycle to reduce the effects of laser diode drift. This test achieved the best accuracy and is also quite indicative of

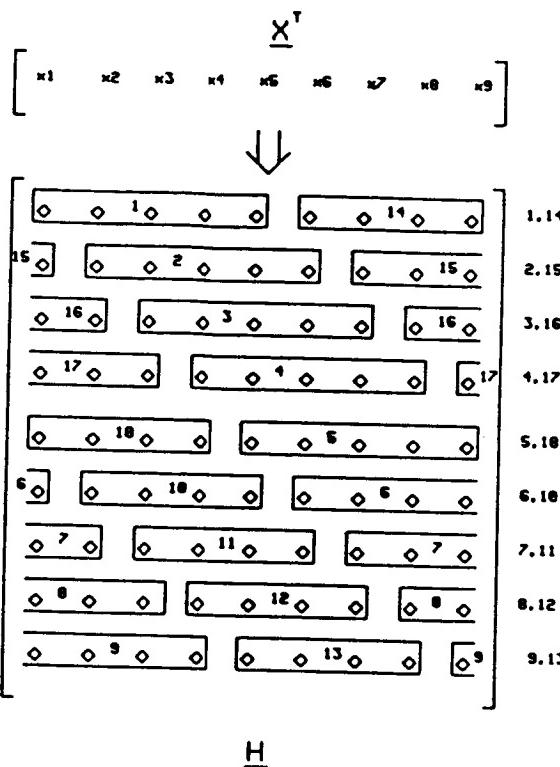
ORIGINAL PAGE IS  
OF POOR QUALITY



**Figure 3-5:** Photograph of the electronic support system.

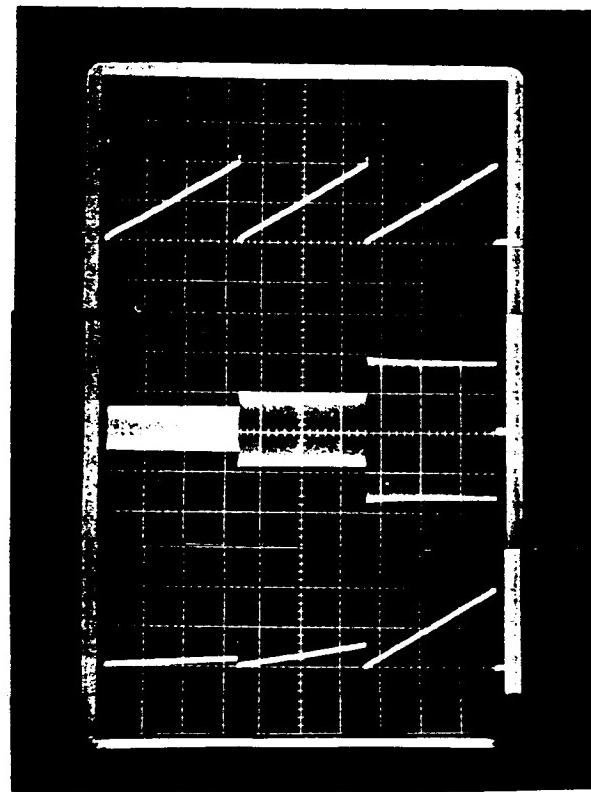
the performance that one can obtain with better temperature stabilization employed. The flexibility of this system and our electronic support hardware make such tests possible. Test 3 indicates the performance obtained with frequency-multiplexing. It shows negligible degradation from the results in Test 1. Test 4 is a simulated result with error source models for all components included in the simulation. Its agreement with laboratory tests indicates the validity of our simulator and error source model.

Many applications exist for such processors in areas such as: optical artificial intelligence, associative memory processors, hypothesis testing systems<sup>12</sup> and for optical interconnections. In Fig. 3-9 we show one architecture suitable for interconnecting  $N$  inputs fed to  $P_1$  with  $N$  outputs at  $P_3$ . In this and similar advanced cases, multi-channel AO cells are employed at  $P_2$ . With the proper frequency fed to the different channels of a multi-channel cell at  $P_2$  of Fig. 3-9,



**Figure 3-6: Data flow arrangement for partitioning**

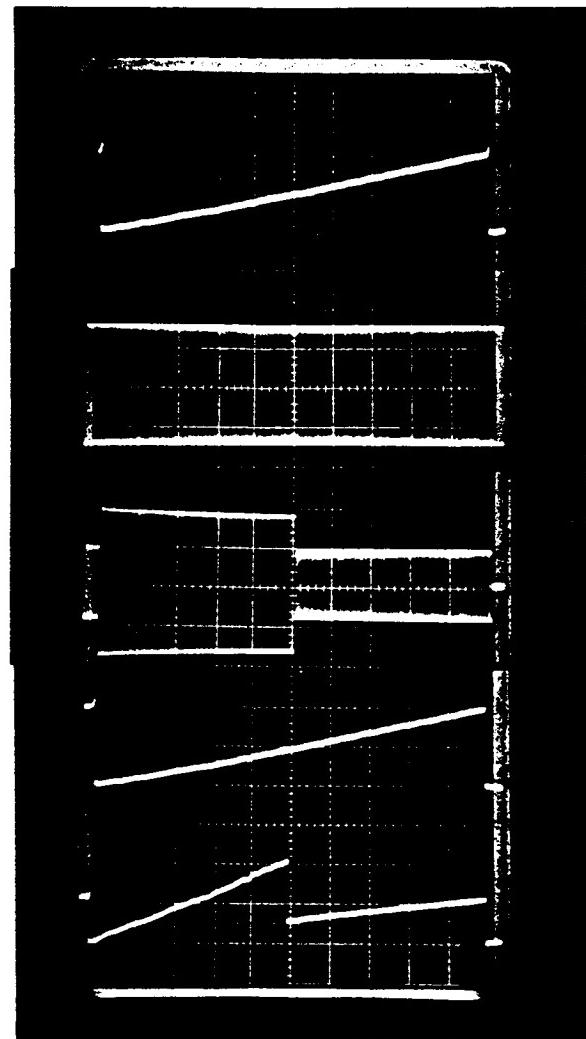
any of the  $P_1$  inputs can be connected to any of the  $P_3$  outputs. This architecture of Fig. 3-9 is due independently to various authors. If all  $N$  inputs are the same, then the system can operate in a broadcasting mode as would be needed for clocking and similar operations. Many useful architectures and algorithms are thus possible with a basic space integrating frequency-multiplexed matrix-vector processor, especially when multi-channel AO cells are included. If the full length of the multi-channel AO cell in Fig. 3-9 is employed, then one can envision using this dimension of the system to encode data, thus achieving both high performance (number of multiplications per second) and high accuracy (with advanced encoding techniques).



**Figure 3-7:** Linear analog performance of 10 bits (0.1%).

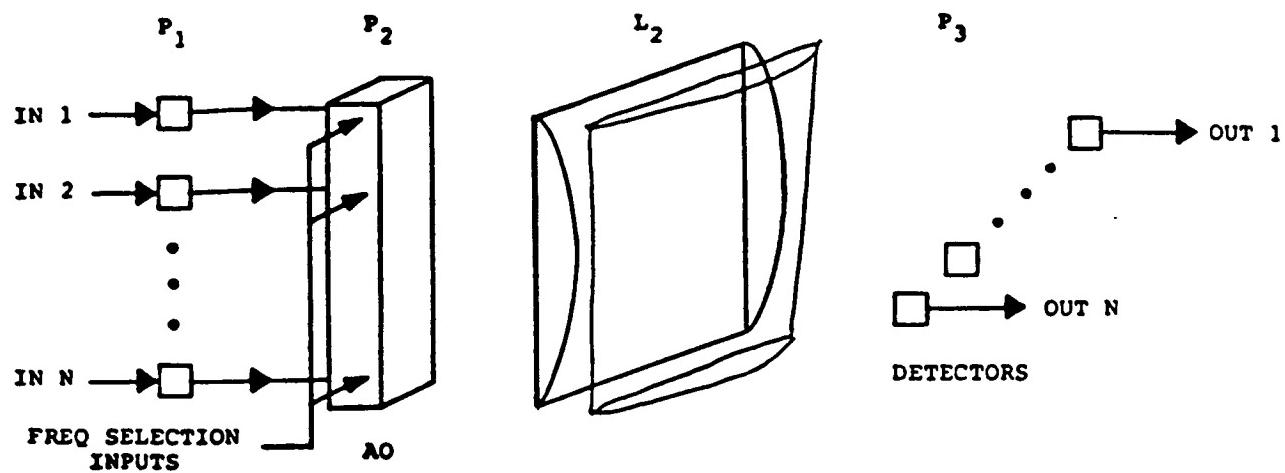
### 3.6 Summary and Conclusion

We have advanced a new space and frequency-multiplexed architecture for matrix-vector processing. We have also noted several new partitioning methods for data in such a processor. The on-line electronic support system for such a flexible (analog and high-accuracy) optical linear algebra processor has been detailed and demonstrated. The major accomplishment has been the demonstration of the solution of a real world problem on such an optical matrix-vector processor.



**Figure 3-8:** Linear analog frequency-multiplexed performance.

ORIGINAL PAGE IS  
OF POOR QUALITY.



**Figure 3-9:** Frequency-multiplexed optical interconnection architecture.

## 4. REAL-TIME OPTICAL LABORATORY LINEAR ALGEBRA SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

### **4.1 Introduction**

A Space Integrating (SI) Optical Linear Algebra Processor (OLAP) employing space and frequency-multiplexing, new partitioning and data flow, and achieving high accuracy performance with a non base-2 number system is described. Laboratory data on the performance of this system and the solution of parabolic Partial Differential Equations (PDEs) is provided. A multi-processor OLAP system is also described for the first time. Its use in the solution of multiple banded matrices that frequently arise is then discussed. The utility and flexibility of this processor compared to digital systolic architectures should be apparent.

Many OLAPs have been suggested<sup>7</sup>, but few have been fabricated and limited laboratory use of these systems in the solution of practical engineering problems has been presented<sup>13, 14</sup>. In Section 4.2, we review one well-engineered OLAP architecture and discuss its laboratory fabrication, its electronic support system and its performance. In Section 4.3, we discuss several features and uses of the system to demonstrate its versatility. Our case study and the algorithm are then detailed in Sections 4.4 and 4.5. Optical realization issues are discussed in Section 4.6 and laboratory results are then advanced in Section 4.7.

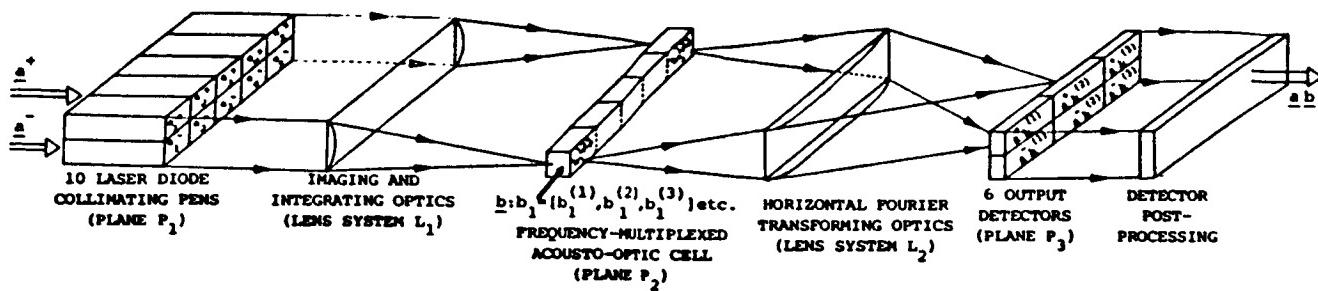
### **4.2 Olap Architecture and Fabrication**

The OLAP we consider is shown in Figure 4-1. Plane  $P_1$  is imaged onto  $P_2$  and the output light leaving  $P_2$  is space integrated onto  $P_3$ . Multiple linear point modulator arrays at  $P_1$  are used to allow bipolar (using two linear  $P_1$  arrays) and complex-valued (using three linear  $P_1$  arrays) data to be processed. Frequency-multiplexing at  $P_2$  (using two or three frequencies) is used to achieve bipolar and complex-valued  $P_2$  data. The input  $P_1$  vector data multiplies the

multiple vector data at  $P_2$  and the Vector Inner Product (VIP) outputs appear on separate horizontal detectors at  $P_3$ . The VIPs of different input  $P_1$  data appear at different vertical locations in  $P_3$ . This space and frequency multiplexed SI OLAP employs local and global interconnections.

The system is fabricated using Laser Diodes (LDs) with individual collimating optics for each  $P_1$  point modulator and a  $\text{TeO}_2$  Acousto Optic (AO) cell at  $P_2$  with a  $T_A = 1 \mu\text{s}$  aperture time, a bandwidth  $BW_A = 200 \text{ MHz}$  and a center frequency  $f_c = 200 \text{ MHz}$ . Three output  $P_3$  detectors are fiber optically coupled to Selfoc lenses in the detector plane to accommodate adequate spacing of detectors. We denote the temporal separation between data packets in  $P_2$  (the different  $P_2$  regions illuminated by different  $P_1$  point modulators) by  $T_B$ . At 10 MHz operation ( $T_B = 0.1 \mu\text{s}$ ), the present system supports  $N = 10$  point modulators and achieves 200 MOPs (millions of operations per second, where an operation is a complex-valued multiplication and addition). The present laboratory data is taken with a 4 MHz data rate per channel ( $T_B = 250 \text{ ns}$ ) on a system using 5 input LDs at  $P_1$ .

The electronic support system for this processor was detailed elsewhere<sup>5</sup>. It includes parallel high-speed memory channels with 12 parallel output bits per channel each at 10 MHz. Each parallel output memory channel is fed to a D/A and to one of the  $P_1$  and  $P_2$  inputs. The  $P_3$  detector outputs are fed through parallel A/Ds to parallel input memory channels. The entire system is under control of an 68,000-based microprocessor with tape, disc, terminal, monitor, etc. and with a VAX port. The entire system is thus quite self-contained and well-engineered. This is essential to allow quantitative data to be obtained and to guide future research and OLAP design.



**Figure 4-1:** Simplified schematic of the space integrating optical linear algebra processor.

### 4.3 System Properties and Use

The architecture of Fig. 4-1 is very versatile. It can operate linear (analog) to 9-10 bits of accuracy. This is achieved by RAM correction of spatial bias and gain variations in the  $P_1$  point modulators and the  $P_3$  detectors, as well as correction of spatial variations in attenuation and response of the AO cell and the frequency response variations of the AO cell (which transfer to  $P_3$  errors, because of the output Fourier transform formed between  $P_2$  and  $P_3$ ). Temporal settling time errors, random time-varying noise, and AO cell dispersion are the major non-correctable errors. All spatial and fixed errors are correctable<sup>15</sup>. Bipolar and complex-valued data can be represented by space-multiplexing at  $P_1$  and frequency-multiplexing at  $P_2$ . Time-multiplexing is also possible and has the advantage that it cancels  $P_1$ ,  $P_2$  and  $P_3$  biases.

The same system can also achieve high-accuracy performance using encoded data. To multiply two encoded numbers, the Digital Multiplication by Analog Convolution (DMAC) algorithm<sup>9, 10</sup> is used. This involves the convolution of the two encoded data schemes achieved with one word fixed at  $P_1$  and the second word fed to  $P_2$ . This yields a mixed radix output. It is converted to conventional binary notation by A/D converting each output bit, shifting it and adding it to the next bit. This same DMAC algorithm operates with data encoded in any base  $B^6$ . With  $D$  digits of data (e.g.  $D = 5$  point modulators at  $P_1$ ). We achieve a dynamic range of  $(B-1)^D$ . One can also represent bipolar data in DMAC by operating the system in a negative base<sup>4</sup>. Thus, this is a most versatile system.

Many data flow and partitioning methods are possible on this system. Consider a matrix-vector multiplication. One can feed the matrix diagonals time-history to different  $P_1$  point modulators and the vector data to the AO cell. This allows partitioning along the matrix diagonals and is quite suitable for banded matrices. When the vector is longer than the AO cell's number of data slots, only part of the vector is present in  $P_2$  at any  $T_B$  and a different part of the vector is present in each  $T_B$ . During each  $T_B$ , the associated  $N$  elements of the matrix are easily determined and fed in parallel to  $P_1$  ( $N$  elements each  $T_B$ )<sup>13</sup>. This is the partitioning method we used in our earlier demonstration of the use of this system in the solution of nonlinear matrix equations, specifically the algebraic Riccati equation<sup>13</sup>. For matrices with multiple bands (e.g. banded matrices in which one band is separated from the other by many elements, as arises in PDEs), the non-zero matrix elements on each row can be fed to  $P_2$  and repeated with the associated required vector elements easily determined and fed in parallel to  $P_1$  each  $T_B$ . This method will be used in our PDE case study. Another new partitioning method that improves throughput involves feeding successive encoded numbers to  $P_2$  on separate frequencies time-multiplexed. This avoids dead time in loading and unloading the AO cell and improves performance by a factor of 1.8. This frequency-multiplexed operation is included in our laboratory experimental data results.

#### 4.4 Problem Definition

We consider the solution of a parabolic PDE on an analog and a high-accuracy version of the same OLAP laboratory system. The specific parabolic PDE selected is the transient diffusion equation with two spatial variables plus time,

$$u_t = \alpha(u_{xx} + u_{yy}), \quad (4.1)$$

where subscripts denote partial derivatives with respect to time, x, or y (e.g.  $u_{xx}$  denotes the second partial derivative). The objective is to determine the temperature distribution  $u$  as a function of space (x,y) and time t. We consider the case when the thermal diffusivity  $\alpha$  is

constant, which is typical of an isotropic time-variant medium. The extension to the non-isotropic case is straightforward with  $\alpha$  becoming a function of the spatial coordinates ( $x, y$ ). The temporal evolving nature of the problem with time requires solutions  $u(x, y, t_n)$  at different time instances to be used to calculate the solution  $u(x, y, t_{n+1})$  at the next time instant. The two types of problem formulations and solutions are explicit matrix-vector (M-V) and implicit LAE (Linear Algebraic Equation) formulations. Both begin with a finite difference solution to (4.1) with a forward difference (forward Euler) approximation of  $u_t$  as

$$(u_{ij}^{n+1} - u_{ij}^n)/\Delta t,$$

where  $n$  is the time index and  $(i, j)$  are the space indices, i.e.  $u_{ij}^{n+1}$  is  $u[i\Delta x, j\Delta y, (n+1)\Delta t]$ , where  $\Delta x$ ,  $\Delta y$  and  $\Delta t$  are the space and time step sizes used. At each spatial location  $(i, j)$ , the temperature at successive times  $n$  and  $n+1$  are calculated and differenced to produce  $u_t$ .

In the explicit solution, we approximate  $u_{xx}$  by a double central difference in  $x$  (index  $i$ )

$$u_{xx} = [u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n]/(\Delta x)^2. \quad (4.2)$$

A similar approximation is made for  $u_{yy}$ .

#### 4.4.1 Explicit 1-D M-V Solution

For the 1-D problem

$$u_t = \alpha u_{xx}, \quad (4.3)$$

this yields

$$u_i^{n+1} = \lambda u_{i+1}^n + (1-2\lambda)u_i^n + \lambda u_{i-1}^n, \quad (4.4)$$

where

$$\lambda = \alpha \Delta t / (\Delta x)^2. \quad (4.5)$$

This shows that the temperature at time step  $n+1$  appears explicitly on the left hand side in terms of constants and spatial solutions at the prior time steps  $n$ . If we denote the spatial solutions for all  $i$  at time  $n$  by  $\underline{u}^n$ , then a M-V description of (4.4) results:

$$\underline{u}^{n+1} = \underline{\mathbf{A}} \underline{u}^n, \quad (4.6)$$

where the matrix  $\underline{\mathbf{A}}$  has  $(1-2\lambda)$  for all main diagonal elements and  $\lambda$  for elements of the diagonals above and below the main diagonal,

$$\underline{A} = \begin{bmatrix} (1-2\lambda) & \lambda & & & \\ \lambda & (1-2\lambda) & \lambda & & \\ & \lambda & (1-2\lambda) & * & \\ & & \lambda & * & * \\ & & & * & * \\ & & & * & * \\ & & & & \lambda \\ & & & & \lambda & (1-2\lambda) \end{bmatrix} \quad (4.7)$$

The explicit solution thus allows us to obtain the temperature distribution at any time  $n\Delta t$  by an M-V multiplication involving the spatial temperature solutions at the prior time  $(n-1)\Delta t$ . The matrix in this 1-D case is banded with a bandwidth of 3.

#### 4.4.2 Implicit LAE Solution

In the implicit solution,  $u_{xx}$  is approximated by the average of (4.2) at  $(n+1)\Delta t$  and  $n\Delta t$ . This yields the Crank Nicholson implicit formulation<sup>16</sup>

$$\underline{B}_1 \underline{u}^{n+1} = \underline{B}_2 \underline{u}^n \quad (4.8)$$

where the matrices  $\underline{B}$  in (4.8) are also banded. A similar approximation is made for the derivative  $u_{yy}$  in the 2-D problem. The implicit solution in (4.8) requires the solution of a M-V equation, i.e. the solution of a set of Linear Algebraic Equations (LAEs) at each time step  $n\Delta t$ . This is much more computationally burdensome than the simpler banded M-V multiplication required in (4.6) at each time step. One can calculate  $\underline{B}_1^{-1}$  in advance and solve (4.8) explicitly in the form  $\underline{u}_{n+1} = \underline{B}_1^{-1} \underline{B}_2 \underline{u}^n$ . However, the matrix is not banded now and hence calculations are significantly complicated with a full matrix present. The approximation in (4.4) is stable for  $\lambda \leq 0.5$  and thus for  $\Delta x$  fixed, small time steps  $\Delta t$  are required and hence many time steps and many M-V multiplications can be required. The approximation in (4.8) yields more exact results and is stable for all  $\Delta t$ ,  $\Delta x$  and  $\Delta y$  values. We discuss computational error effects (as distinguished from algorithmic accuracy issues) associated with this algorithm in a later section, as well as the use of the implicit LAE algorithm with different  $\Delta$  steps.

#### 4.4.3 Explicit Matrix-Vector 2-D Solution

For the explicit solution to the 2-D problem in (4.1), the finite difference approximation yields

$$u_{ij}^{n+1} = \lambda u_{i+1,j}^n + \lambda u_{i-1,j}^n + \lambda u_{i,j+1}^n + \lambda_{i,j-1}^n + (1-4\lambda)u_{ij}^n. \quad (4.9)$$

To obtain a M-V problem<sup>17, 18</sup>, we order the  $N^2$  elements of  $u_{ij}$  into an  $N^2$  vector  $\underline{u} = [u_{11}, \dots, u_{NN}]^T$ . With  $\Delta x = \Delta y$ , this yields the M-V explicit solution of the form of (4.6) with the matrix  $\underline{\underline{A}}$  having the central three diagonal elements non-zero and two other non-zero diagonals  $N-1$  elements away from the main diagonal. With other high-order difference approximations, more non-zero diagonals  $2N-1$  away from the main diagonal result. We do not consider such cases, since the resultant problem presently under consideration suffices and can be generalized to other problems as they arise.

If we renumber the grid point elements, the bandwidth of the matrix will decrease, however algorithms to renumber nodes are quite time-consuming. Our proposed multi-processor and other architectures are appropriate for the simplest node numbering method employed. The form of the 2-D implicit solution is the same as in (4.8) with the same double-banded matrix structure existing as occurred in the explicit solution of an LAE required at each time step. The need to utilize and preserve the banded nature of the matrix now becomes of more concern. If a direct LAE solution were used, all central  $2n+1$  diagonals would fill in and become non-zero. This significantly increases the number of matrix multiplications required and the size of each. If iterative LAE solutions are used, the number of iterations required (each involving a M-V multiplication) is difficult to calculate, although estimates of it are possible<sup>19</sup>. For an implicit solution, an iterative LAE solution is preferable, in general. Thus, for these reasons, the explicit solution in (4.8) and (4.9) was chosen for implementation on our laboratory system. The boundary conditions for the matrix must still be included in our problem formulation. This is detailed in Section 4.6. However, the general matrix structure and the size of the matrix is as

described above (i.e. an  $N^2 \times N^2$  matrix A with multiple bands separated by  $N$  elements and with  $N^2$  vectors u).

## 4.5 Case Study

The case study chosen was the solution of the 2-D diffusion equation for a  $10 \times 10 \text{ cm}^2$  square aluminum plate with thermal diffusivity  $\alpha = 0.86 \text{ cm}^2/\text{sec}$  for the case when the plate is divided into a uniform grid of  $N \times N = 11 \times 11 = 121 = N^2$  square elements each  $1.0 \times 1.0 \text{ cm}^2$  (i.e.  $\Delta x = \Delta y = 1.0 \text{ cm}$ ) and with boundary conditions of zero temperature for the 40 boundary points on the edges of the plate. To satisfy  $\lambda \leq 0.5$ , we require time steps  $\Delta t \leq 0.29 \text{ sec}$ . At each time step, we calculate  $u(x,y,t)$  at the 81 interior points on the grid. We used a natural ordering of the grid points on the 2-D plate from left-to-right and top-to-bottom (e.g.  $u_{12} = u_{i,j}$  is the first interior element in the second row). The matrix in our 2-D problem is  $N^2 \times N^2 = 121 \times 121$ . We note that  $4N-4$  of the rows of this matrix are altered by the boundary conditions associated with the  $4N-4$  edge elements. The full matrix consists of  $N \times N$  blocks each with  $N \times N$  elements (see Fig. 4-2). The top left block and bottom right block are the identity matrix I, since the first and last rows of grid elements are edge elements always clamped to zero temperature (i.e.  $u^n = u^{n+1}$  for these elements). The remaining elements in these rows are zero. The structure for the other diagonal blocks are all similar. They are all tri-diagonal except for the first and last rows of each block which are zero except for a "1" on the diagonal. All other elements of these rows are zero. The blocks removed by one element from the main diagonal have  $\lambda$  along the diagonal (except for the rows noted above). The remaining elements of the matrix are zero.

Our case study thus involves the solution of

$$\underline{u}^{n+1} = \underline{A} \underline{u}^n, \quad (4.10)$$

where u is a  $N^2$  vector and the matrix A is an  $N^2 \times N^2$  matrix as shown in Fig. 4-2, with

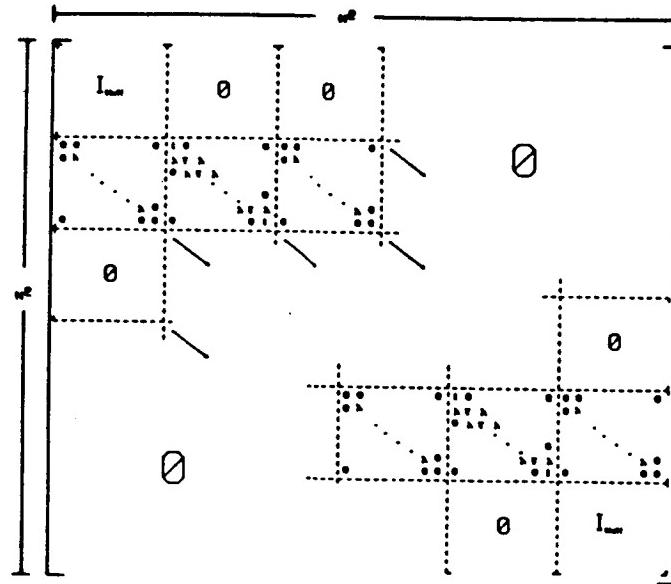
$$\lambda = \gamma \Delta t / (\Delta x)^2, \quad \gamma = 1-4\lambda \quad (4.11)$$

*Boundary Conditions = 0 Temperature on Edges* (4.12)

$\Delta t = 0.29 \text{ sec}$  (4.13)

*Initial Temperature (interior points) =  $u(x,y,0) = 1$*  (4.14)

$N^2 = 11^2 = 121 \text{ Grid Point Elements.}$  (4.15)



**Figure 4-2:** Structure for the matrix in the implicit 2-D diffusion equation solution

The temperature  $u = u_{ij}$  at all 81 interior points is calculated each  $\Delta t$  using (4.10) on the optical laboratory system. To calculate  $u_{ij}$  for each of the 81 interior points requires a VIP, i.e. 81 bipolar VIPs or  $81 \times 4 = 324$  unipolar VIPs. Each VIP requires five multiplications and additions and each is achieved with  $2N-1 = 9$  convolutions in a high-accuracy encoded algorithm. Each convolution requires five multiplications and additions. The total number of multiplications/additions for 50 time steps is thus  $324 \times 5 \times 50 \times 9 \times 5 = 3.6 \text{ million}$ . As we shall see, these operations were all performed to sufficiently high-accuracy with no errors on the laboratory system at a rate of five multiplications/additions per  $T_B = 250 \text{ ns}$ .

## 4.6 Optical Realization Issues

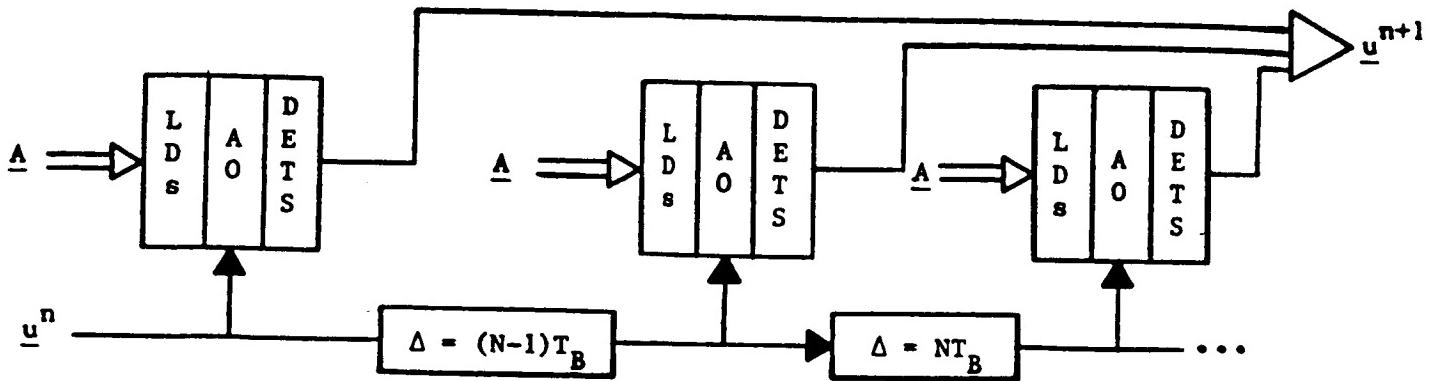
### 4.6.1 Node Numbering

We retain the conventional grid point numbering to allow different solutions and boundary conditions to be considered, rather than requiring new optimum node numbering techniques for each problem. A considerable amount of effort can arise in the node numbering phase of problem formulation and to avoid this and to concentrate on problem solutions, we consider only conventional left-to-right and top-to-bottom node numbering. In general, this results in the central diagonal block matrices being diagonal (their bandwidth equals 3 in our case) with other non-zero elements being separated from the main diagonal by  $\pm(N+1)$  elements. For higher-order differencing schemes, other non-zero elements and block diagonal matrices will exist at  $2N$  etc. elements from the main diagonal.

### 4.6.2 Partitioning and Data Flow

These issues address how the matrix and vector data is fed to the  $P_1$  and  $P_2$  data planes of the system and how the  $P_3$  processing required to obtain the final desired output is obtained. Fig. 4-3 shows the three processor scheme with the non-zero matrix diagonal time-history fed to  $P_1$  and delayed versions of  $\underline{u}$  fed to subsequent processors. In general, each  $P_1$  data plane requires  $M$  point modulators where  $M$  is the largest bandwidth for any block matrix (this is 3 in our case), the number of processor equals the number of block matrix bands (3 in our case), and the delays are as shown in the figure. These delays arise because there are many non-zero elements between and separating the different bands of the matrix (in our case, these bands are separated by approximately  $N$  grid points). The  $P_3$  outputs are then the desired  $N^2$  elements of the temperature vector  $\underline{u}$  at each  $n\Delta t$  time step.

A multiple banded matrix problem can also be solved on a single processor with the matrix fed to  $P_1$  and the vector  $\underline{u}$  fed to  $P_2$ . However, this requires that the number of  $P_1$  point



**Figure 4-3:** Multi-processor architecture for multiple banded matrices

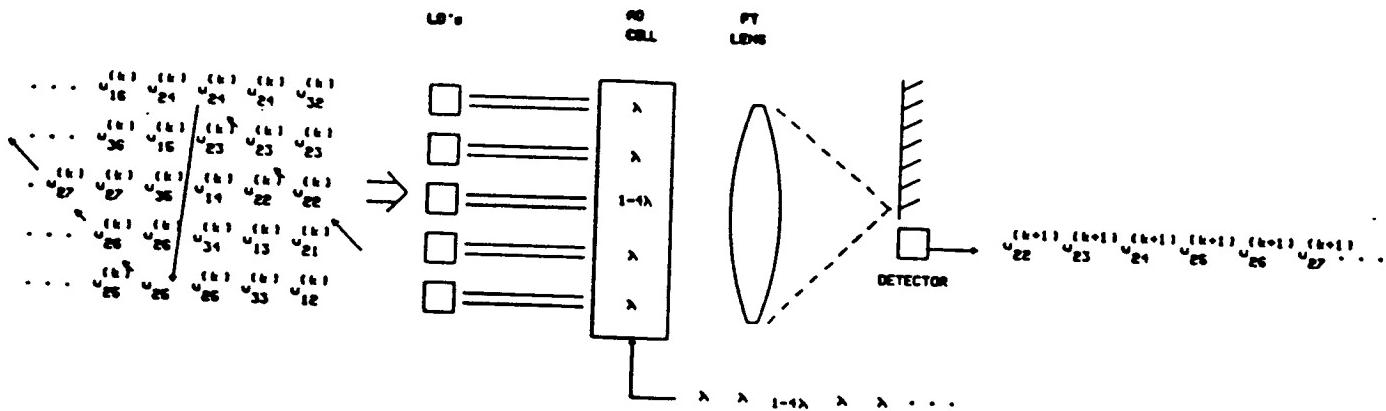
modulators equal the total number of non-zero matrix diagonals. The number of time slots used in the AO cell also equal this. This is only 5 rather than 3 for our present case, however in other problems this number will be considerably larger. However, the AO cell must now support  $2N+1$  time increments of the vector data and the associated  $T_A = (2N+1)T_B$  length of the AO cell is not attractive and is wasteful of the AO cell's TBWP<sub>A</sub>. This requirement arises because at any instant, the five point modulators at P<sub>1</sub> must interact with the associated  $\underline{u}$  elements n, N+n, (N+1)+n, (N+2)+n, and (2N+1)+n along the AO cell. This arises since the 3 matrix bands are separated by N-2 elements for a grid with N points in 1-D.

An alternate and preferable single processor data flow arrangement (and the one we implemented in our laboratory system) involves feeding the matrix elements to the AO cell at P<sub>2</sub> and the temperature vector data  $\underline{u}^n$  to the P<sub>1</sub> point modulators. This still requires only five P<sub>1</sub> point modulators and now an AO cell with only  $T_A = 5T_B$  length. The five matrix elements and present in the AO cell are always the same ( $\lambda, \lambda, 1-4\lambda, \lambda, \lambda$ ) and this five elements problem is cyclically repeated continuously. The five P<sub>1</sub> point modulators are fed with the associated  $\underline{u}_n$  elements  $\underline{u}_{i,j}^n$  required. To calculate  $\underline{u}_{2,2}^{n+1}$  (the first interior element), the implicit algorithm in (4.9) requires the five  $\underline{u}^n$  input elements shown below:

$$\underline{u}_{2,2}^{n+1} = \lambda \underline{u}_{2,1}^n + \lambda \underline{u}_{3,2}^n + (1-4\lambda) \underline{u}_{2,2}^n + \lambda \underline{u}_{2,3}^n + \lambda \underline{u}_{1,2}^n \quad (4.16)$$

The general ordering of the five  $\underline{u}^n$  elements required at P<sub>1</sub> at each  $T_B$  follows from the

regularity of the pattern in (4.16) for subsequent interior elements. These  $P_1$  point modulators to which the five  $\underline{u}^n$  elements are fed is orchestrated in conjunction with the movement of the five matrix elements through the AO cell. It is easy to show that these arrangement achieves an entire set of  $N^2 \times N^2$  temperature updates  $\underline{u}(x,y)$  at a given time in  $(N-2)^2 + (M-1)$  computational intervals  $T_B$ . This is the minimum number of non-zero multiplications possible, where  $N$  is the number of points in 1-D in the grid and  $N-2$  is the number of non-zero interior points in 1-D, and  $M$  is the number of non-zero diagonals in 1-D in the matrix. The second  $(N-1)$  term is the start-up interval to first load the AO cell and it is negligible. Figure 4-4 shows the input data sequence for calculation of the second row of interior elements  $u_{2,2}$  to  $u_{2,9}$  at time  $(k+1)\Delta t$ .



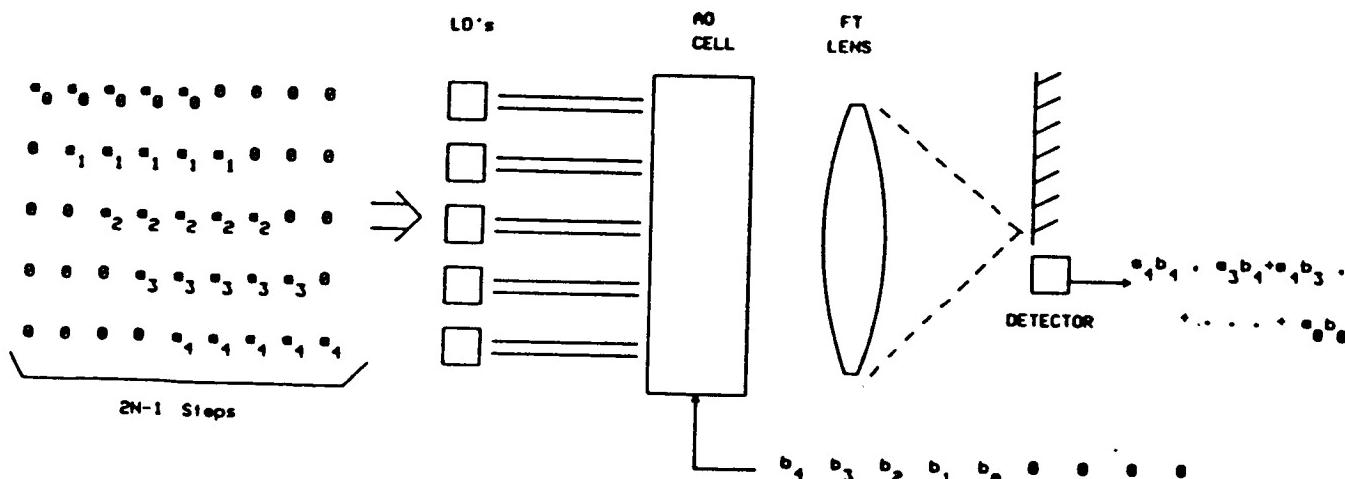
**Figure 4-4:** Data sequence for updating of explicit M-V formulation of 2-D diffusion PDE

#### 4.6.3 Partitioning

For cases when the number of non-zero diagonals exceeds the number of  $P_1$  point modulators, we can extend the diagonal partitioning to subsequent time steps on one processor and can assemble the results appropriately delayed at  $P_3$  is detailed elsewhere<sup>6</sup>. The data flow in Fig. 4-4 can similarly be partitioned.

#### 4.6.4 High-Accuracy Encoding

With the DMAC algorithm and base B, the system of Fig. 4-1 can achieve high-accuracy calculations. This is required in the case study under consideration, because of the cumulative effects of errors in the  $\underline{u}^n$  calculated at  $n\Delta t$  propagate to the  $\underline{u}^{n+1}$  values calculated at  $(n+1)\Delta t$ . These remarks apply to all implicit and explicit solutions since all are open-loop algorithms that extrapolate to the next time step based upon calculations from the prior time step. With five  $P_1$  point modulators, we achieve a dynamic range of  $(B-1)^5$  for our calculations. Our laboratory data used  $B=5$  and achieved a  $4^5 \approx 15$  bit =  $2^{15}$  computational precision. This is sufficient to demonstrate the point in concept. The data flow for the high-accuracy multiplication of two 5-digit numbers  $a_0 \dots a_4$  and  $b_0 \dots b_4$  is shown in Fig. 4-5. As seen, each LD output is fixed for  $5T_B$  with each input  $A$  data streams skewed by  $1T_B$ . The  $2N-1 = 9$  digit output data stream is obtained with the MSB  $a_0 b_0$  produced first in the laboratory realization shown and used. This allows round-off or termination after 5 calculated digits if desired.



**Figure 4-5:** Data flow for the high-accuracy multiplication  
(the example shown is for the product of two 5-digit numbers)

#### **4.6.5 High-Accuracy Data Flow and Partitioning**

To simplify data flow and to reduce the number of high-accuracy multiplications required to a minimum, we calculated  $\lambda \underline{u}^n$  and  $\gamma \underline{u}^n = (1-4\lambda) \underline{u}^n$  for the entire  $N^2$  vector  $\underline{u}^n$  and in  $P_3$  software we assembled the elements  $u_{ij}$  of  $\underline{u}^{n+1}$  using (4.9). This reduced the number of high-accuracy multiplications required to the minimum number possible  $2(N-2)^2$ , where the factor of 2 arises due to the two scalar-vector multiplications used  $\gamma \underline{u}^n$  and  $(1-4\lambda) \underline{u}^n$ . In the laboratory system, the summations of elements in (4.9) was performed after decoding to radix 2. In a real-time system, one would form the sum first and then decode to make the post-processing requirements faster and simpler.

#### **4.6.6 Performance Measures**

To quantify the performance of the laboratory processor, we calculated the exact temperature distribution after different time steps using single precision 24 bit mantissa floating point calculations on a VAX and these results were compared to those obtained on the optical laboratory system. We refer to the digitally calculated results using this method as the ideal results. The maximum percent error in the temperature calculated at any grid point and the average percent error calculated over the plate were determined. These are referred to as the maximum and average error respectively. To pictorially present the results obtained, we displayed the calculated temperature distribution for the  $11 \times 11$  grid in 2-D on a display with white being a temperature of 1 and black being temperature of 0, with 10 increments and steps of 0.1 in the output temperature calculated displayed using different gray levels. Initially, at  $t = 0$ , the interior region of the plate is white and the edges are clamped to 0 (or black). The final steady state temperature of the plate is of course the entire plate being at the edge boundary temperatures of 0 (i.e. black). This 2-D data display is quite useful during laboratory runs to insure that the processor is evolving properly.

## 4.7 Laboratory Test Results

The laboratory system used had only one input  $P_1$  LD array and one set of three output  $P_3$  detectors. The explicit M-V solution was run time-multiplexed with the positive and negative values of the matrix elements fed to the system at separate times and the difference calculated after detection. This operating mode is attractive since  $P_1$ ,  $P_2$  and detector bias effects then tend to cancel.

Simulations were performed to compare the explicit M-V and implicit LAE methods. All laboratory data was obtained using only the explicit M-V solution. In the implicit LAE method, frequency-multiplexing was used to represent the bipolar matrix data and to allow AO cell frequency dispersion effects to be addressed. Since the initial temperature for the interior of the plate was set to 1 (the largest number allowed in the processor), and since the final steady state temperature across the plate was a uniform value of 0 (the lowest number allowed), no data scaling was required and the temperature vector  $\underline{u}$  was unipolar. Thus, the only bipolar data representation of concern was the matrix, not the vector element. In the iterative Richardson algorithm solution used in the implicit LAE method, the acceleration parameter chosen was 0.5. After 10 iterations, the iterative solution error was below the computational errors of the processor. Thus, a fixed number of iterations (10 iterations) and an acceleration factor ( $\omega = 0.5$ ) were used in the implicit LAE algorithm simulations. In both algorithms, the matrix data was fed to the AO cell and was recycled as detailed earlier.

### 4.7.1 Implicit vs. Explicit Solutions with Computational/System Errors Included

The implicit solution is more accurate than the explicit one, because it better approximates the derivative (not because the LAE rather than the M-V algorithm is better). After 20 time steps (5.81 secs), the temperature in the center of the plate was estimated to be 0.5803 by the explicit algorithm, whereas the implicit algorithm yielded a value of 0.5907. The exact value<sup>20</sup>

was 0.593, as obtained from a closed-form solution. Thus, the implicit solution was found to be more accurate than the explicit one. In our simulations, we also included various error sources that can typically be expected in an optical systolic realization. When such error sources are present, we consistently found (from over 20 different simulation runs) that the implicit LAE solution was worse than the explicit M-V one by a factor of 1.4 to 2. This is consistent with a separate theoretical analysis indicating that noise effects will add as the mean square value. Specifically, for noise-like errors in the iterative Richardson algorithm, even when the iterative LAE solution was run until the algorithm errors were below the hardware computational system errors, the Richardson portion of the implicit algorithm was found to add a factor of  $(2)^{1/2} \approx 1.4$  to the noise growth effects of the evolving algorithm. Thus, when system and component errors are included, the explicit algorithm appears to be preferable (both theoretically and from laboratory simulations).

#### **4.7.2 Implicit Algorithm with Variable Time Step Size**

The prior comparisons of the implicit and explicit algorithms used the same  $\Delta t$  time step in both cases, with the choice being made based upon the stability ( $\lambda \leq 0.5$ ) of the explicit algorithm. The time step  $\Delta t$  in the implicit algorithm can be adjusted and with larger  $\Delta t$  steps, fewer iterations will be required and hence the algorithm error would be less. However, when computational errors (such as processor and system accuracy) are included, this tradeoff is not obvious. If the computational errors are large, coarse time steps should improve performance, however with smaller computational errors (as we expect in our system), the algorithmic error can be less with smaller time steps. In tests with a modest amount of system error included in simulations, we found that doubling the time step to  $2\Delta t$  yielded an implicit algorithm average error that was only 60% of the value obtained with a time step of  $\Delta t$ . For larger amounts of computational error, the improvement was less (16%), but still was rather consistent and contributed to polarization errors alone. When the modest computational error case was run

with a step size of  $4\Delta t$ , the average error in the implicit algorithm was found to be 3.8 times worse than that when the step size was  $\Delta t$ . Thus, the use of coarse time steps will not always improve the performance of the implicit algorithm. However, with the proper  $\Delta t$  time step choice, an improvement in performance and optimization is clearly possible.

#### **4.7.3 Analog System Laboratory Performance**

The analog performance of the system is listed in Table 4-1. We do not expect accurate results and thus the purpose of these tests was to quantify and assess different system error sources. As seen, when the output light from the different laser diode input point modulators are isolated more (by reducing their crosstalk), performance improved significantly (compare test 2 to that of test 1 results). In all cases, the temperature distribution was calculated for 50 time steps  $\Delta t$ . The error always increases with time, because of the evolving nature of the algorithm. For an analog system, operation beyond 10 time steps yielded unacceptably large errors above 2%.

#### **4.7.4 Encoded High-Accuracy Laboratory System Performance**

The performance of the system with different bases for the data (column 2) and other cases (column 3) is shown in Table 4-2. With 5 input point modulators or digits, operation in base B yields a dynamic range of  $(B-1)^5$ . As tests 3 and 4 show, no errors were obtained after 40 time steps or  $40\Delta t$  of time, when base 3 and 4 operation was employed. The theoretical probability of error for base 4 operation was theoretically computed by us to be  $4.5 \times 10^{-7}$ . This represents a quite considerably attractive error rate for an optical processor. With base 5 and 6 operation, system noise caused the output data to exceed the separation between levels in the output  $P_3$  A/D data. In the base 5 runs, 3 errors occurred during the  $50\Delta t$  time steps. In a  $50\Delta t$  time step run, 81 grid points are calculated at each time step and thus as noted earlier, a significant number (3.6 million) of multiplications and additions were performed in each algorithm. These operations were performed on the laboratory system with a data rate of 5 multiplications and

**TABLE 4-1:**  
**Optical Laboratory Hardware Results:**  
**Explicit 2-D Transient Diffusion Equation Matrix-Vector Solution (Parabolic PDE)**  
**Analog System Performance Results (4 MHz)**

TEST NUMBER	REMARKS	TEMPERATURE ERRORS			
		AFTER $1\Delta t$ STEP		AFTER $10\Delta t$ STEPS	
		max error	avg error	max error	avg error
1	Low duty cycle (LD temp stabilized)	3.24%	1.86%	20.6%	14.5%
2	low duty cycle and reduced LD crosstalk (alternate LDs)	2.1%	0.62%	7.66%	2.41%

additions per  $T_B = 250$  nsec. Higher performance is possible with more elements in the system, by the use of multi-channel AO cells and with a higher input data rate. The present laboratory system has defined many of these issues and provided very useful initial laboratory results and experience.

In test 7, frequency-multiplexing of the vector elements to the AO cell was used with the matrix elements fixed on the laser diodes. Operation of the frequency-multiplexed system beyond base 3 was not found to be attractive in the present laboratory system, because of spatial variations in the actual AO wave transmitted in the cell at different frequencies. This error source can be corrected with better transducers in the AO cell.

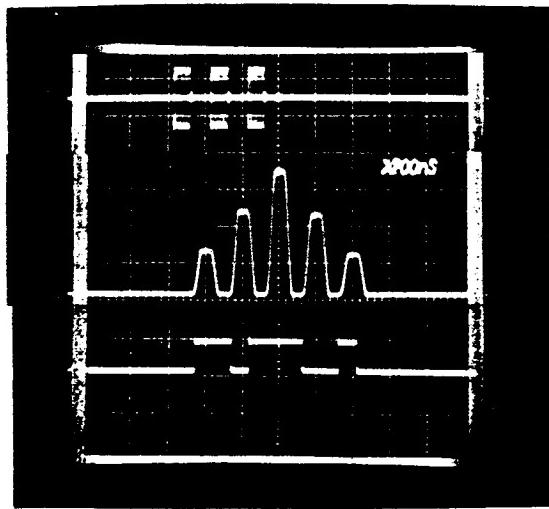
**TABLE 4-2:**  
**Optical Laboratory Hardware Results**  
**Explicit 2-D Transient Diffusion Equation Matrix-Vector Solution (Parabolic PDE)**  
**Encoded High-Accuracy Performance Results (low duty cycle, LDs stabilized, 4 MHz)**  
**5 digits, varying bases B, accuracy =  $(B-1)^5$**

		TEMPERATURE ERRORS									
		after $1\Delta t$		after $10\Delta t$		after $20\Delta t$		after $40\Delta t$			
TEST BASE REMARKS		NO.	USED	max	avg	max	avg	max	avg	max	avg
		3	3	0%	0%	0%	0%	0%	0%	0%	0%
		4	4	0%	0%	0%	0%	0%	0%	0%	0%
		5	5	0%	0%	0%	0%	0%	0%	0.6%	0.02%
<hr/>											
		noise exceeds									
		output A/D									
		6	6	level	0.16%	0.1%	7.7%	1.3%	17.3%	4.3%	
<hr/>											
		separation									
<hr/>											
		frequency									
		7	3	multiplexed	0%	0%	0%	0%	0%	0%	0%

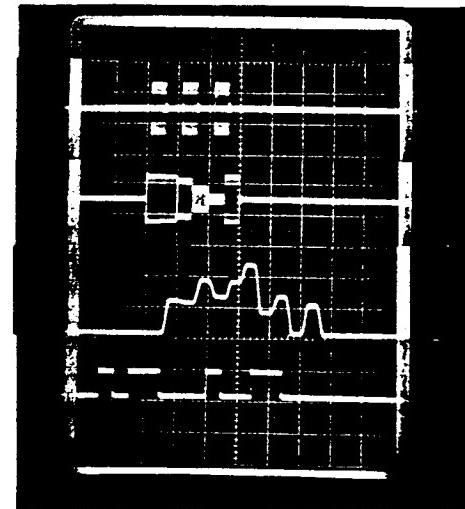
#### 4.7.5 Quantitative Individual High-Accuracy Multiplication Data

To visually show the high-accuracy DMAC performance of the processor, we considered several individual high-accuracy multiplications. Figure 4-6 shows the results for binary encoded data multiplications. The LD input data and RF input data (top trace) are 10101. The mixed-radix detected output obtained on the optical system is the correlation 102030201 as shown in the second trace. The decoded binary output 0110111001 (lower trace) is then obtained by the post-processor in the optical system. In Fig. 4-7, we show results for multi-level encoded data with the laser diode spatial input 10101 on the top trace, the RF time input in base 10 on the second trace (10,9,6,4,1), the mixed-radix output on the third trace and the final decoded binary output on the lower trace.

---



**Figure 4-6:** High-accuracy digital-encoded multiplication numerical example

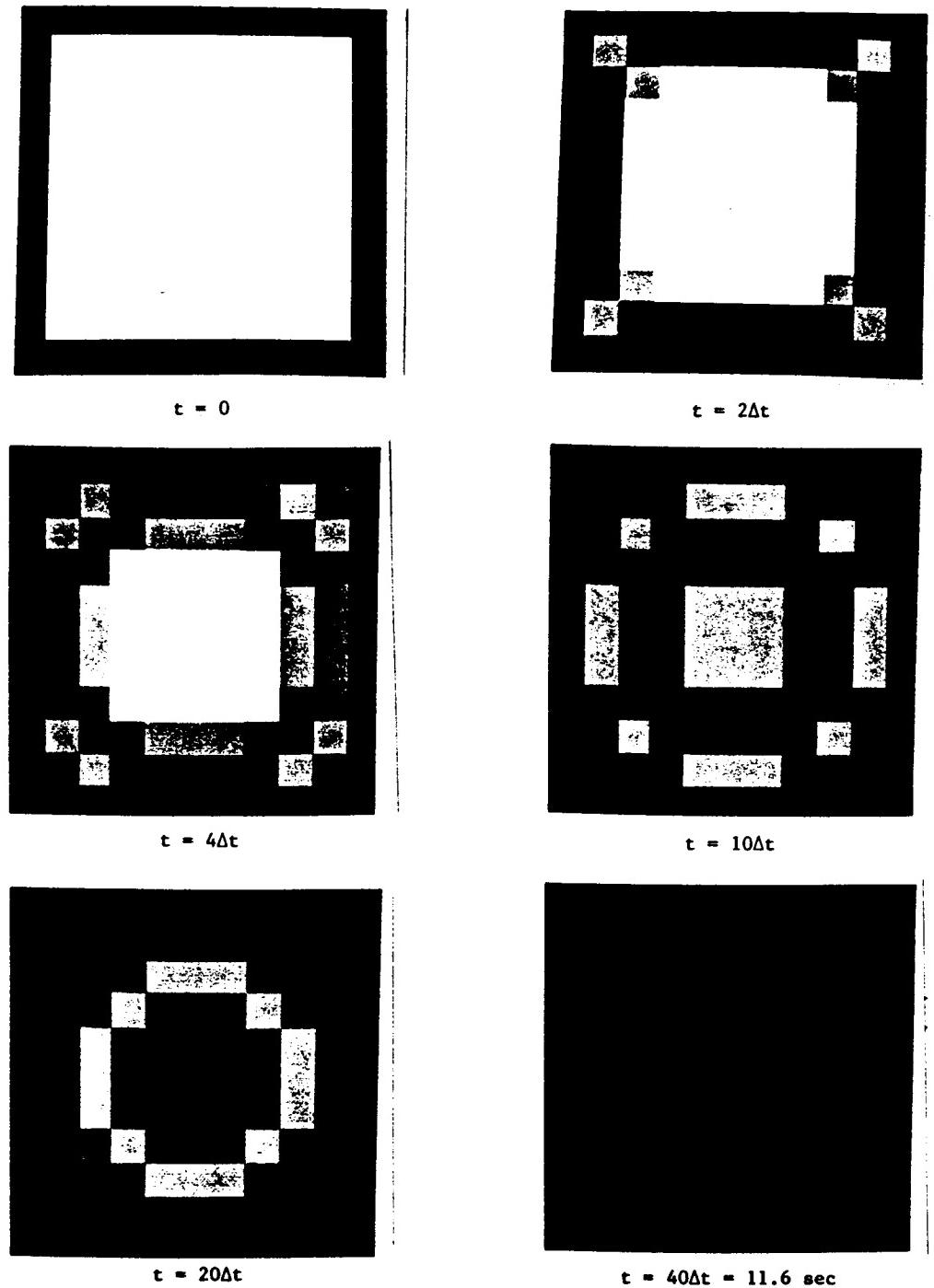


**Figure 4-7:** High-accuracy multi-level encoded multiplication example

ORIGINAL PAGE IS  
OF POOR QUALITY

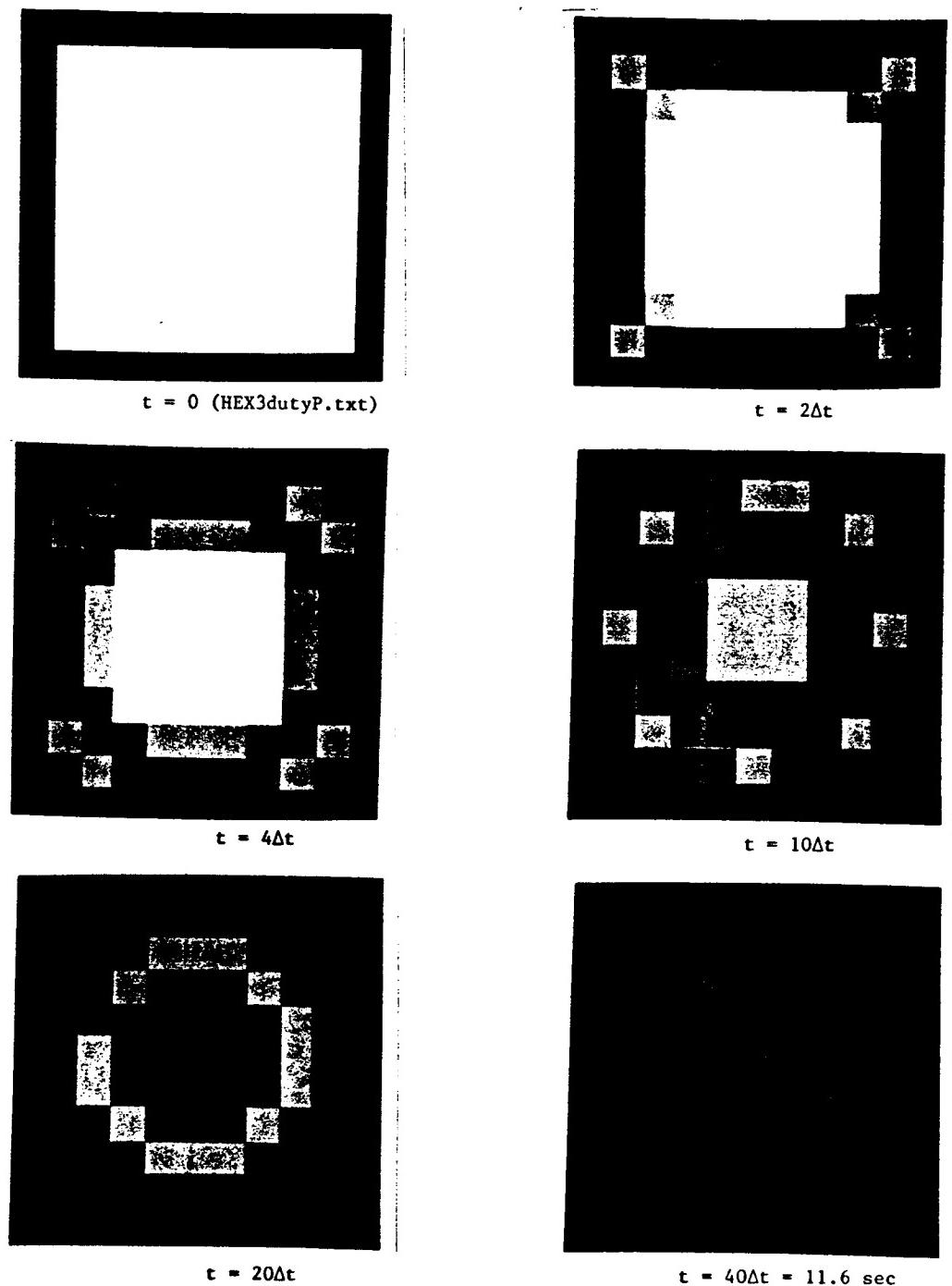
#### 4.7.6 Graphical 2-D Temporal Temperature Data Results

In these 2-D spatial representations of the temperature pattern  $u(x,y)$  at different time increments obtained from the optical processor, we employ the coding method earlier (0 temperature = black, 1 temperature 1 = white, gray-scale encoding is used to represent intermediate temperature values with 0.1 increments being employed). Figure 4-8 shows the theoretically exact output computed on the VAX to single precision. It shows the temperature distribution of the plate's evolution with time from its initial conditions toward its steady state value of 0 temperature across the plate. Figure 4-9 shows the results of our analog computation using reduced crosstalk effects (test 2 in Table 4-1). Its results are quite accurate and clearly map the general trend provided from the theoretical results in Figure 4-8. Figure 4-10 shows the results obtained at different time steps for the case of encoded data in base  $B = 4$ , using 5 channels of the system (i.e. a dynamic range of  $3^5$ ). Figure 4-11 shows the results for the case of an encoded data high-accuracy computation in base  $B = 3$  using 5 channels of the system with frequency-multiplexing employed on the optical laboratory system to achieve higher throughput data results. All of these data results were obtained on the laboratory system of Fig. 4-1 (except for the theoretical data in Fig. 4-8).

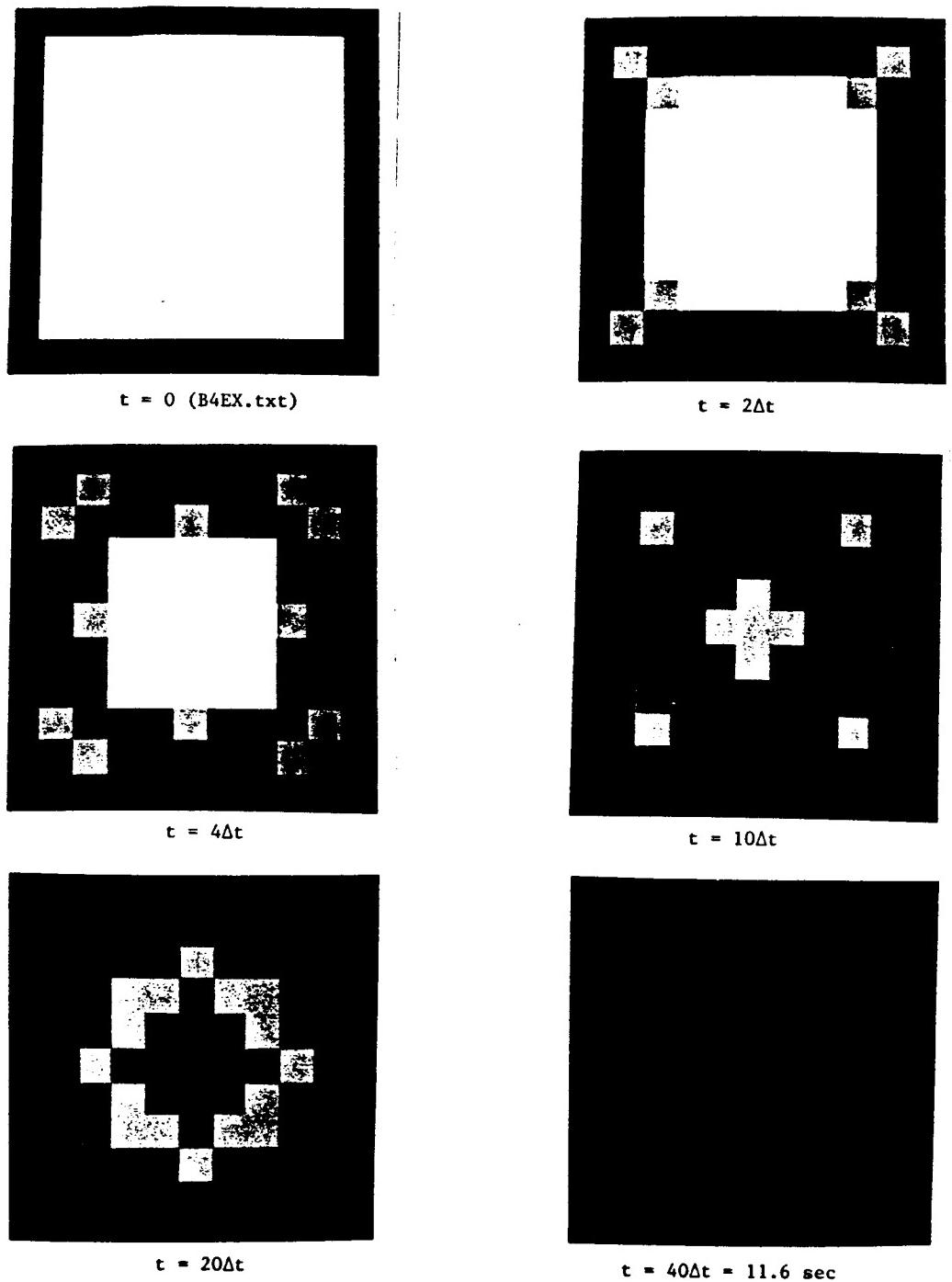


**Figure 4-8:** Theoretical single-precision VAX results expected from the diffusion problem

ORIGINAL PAGE IS  
OF POOR QUALITY.



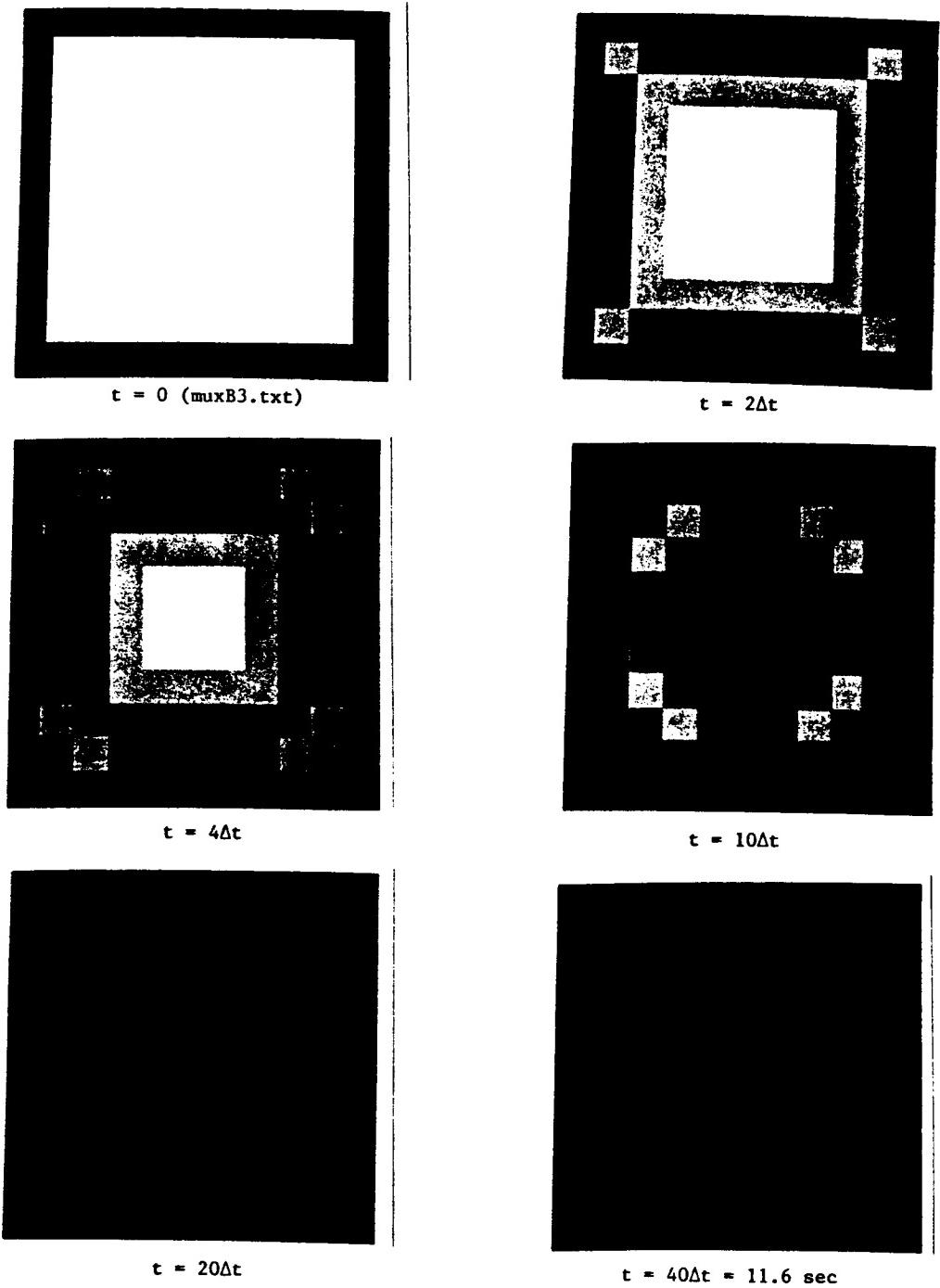
**Figure 4-9:** Laboratory system results obtained with the analog version of the processor with reduced temporal and crosstalk effects of  $P_1$  included



**Figure 4-10:** Optical laboratory test results obtained with base 4 operation with 5 channels of the optical processor

---

ORIGINAL PAGE IS  
OF POOR QUALITY



**Figure 4-11:** Optical laboratory system results obtained with base 3 operation and 5 channels of the optical system employing frequency-multiplexing

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

## 5. TIME AND SPACE INTEGRATING OPTICAL LABORATORY MATRIX-VECTOR ARRAY PROCESSOR

### 5.1 Introduction

The laboratory realization of a hybrid time and space integrating acousto-optic array processor is described with the fabrication of the system and its electronic support and a case study finite element solution on the laboratory system facility emphasized. The output detector system in this processor is unique and allows the use of different number representations. We emphasize the use of this system for a new sign-magnitude bipolar data representation that is quite attractive for use with a new one-channel LU decomposition algorithm and architecture to solve linear algebraic equations (LAEs). These features are employed in our finite element case study. This work represents: the first laboratory optical matrix vector multi-channel processor, the first laboratory realization and demonstration of a new LU decomposition algorithm, the first laboratory LAE direct solution demonstration, the first finite element method optical laboratory solution demonstration, a new mixed-radix to binary conversion technique, plus a new partitioning technique to allow higher accuracy than the number of bit channels permits as well as system hardware and speed trade-offs. Such optical array processors are of use as linear algebra processors, associative memory processors, feature extractors for pattern recognition, and in nearest neighbor classifiers as well as neural networks.

Optical linear algebra processors have received much recent attention<sup>7</sup>, but little laboratory demonstration data<sup>13, 21, 5, 22</sup>. In this paper, we advance the first laboratory realization of a multi-channel matrix vector processor. The basic architecture was previously described in concept<sup>6</sup>. The system architecture is reviewed in Section 5.2. The number representation used [Section 5.3] and algorithm remarks concerning LU decomposition and

obtaining higher accuracy than the number of channels allowed by partitioning, plus other partitioning techniques to solve problems larger than the size of the processor [Section 5.4] are then addressed. The electronic support requirements are summarized in Section 5.5. The electronic laboratory support system and laboratory system fabricated are then summarized in Section 5.6. Our case study is briefly addressed [Section 5.7] and laboratory data obtained is presented [Section 5.8].

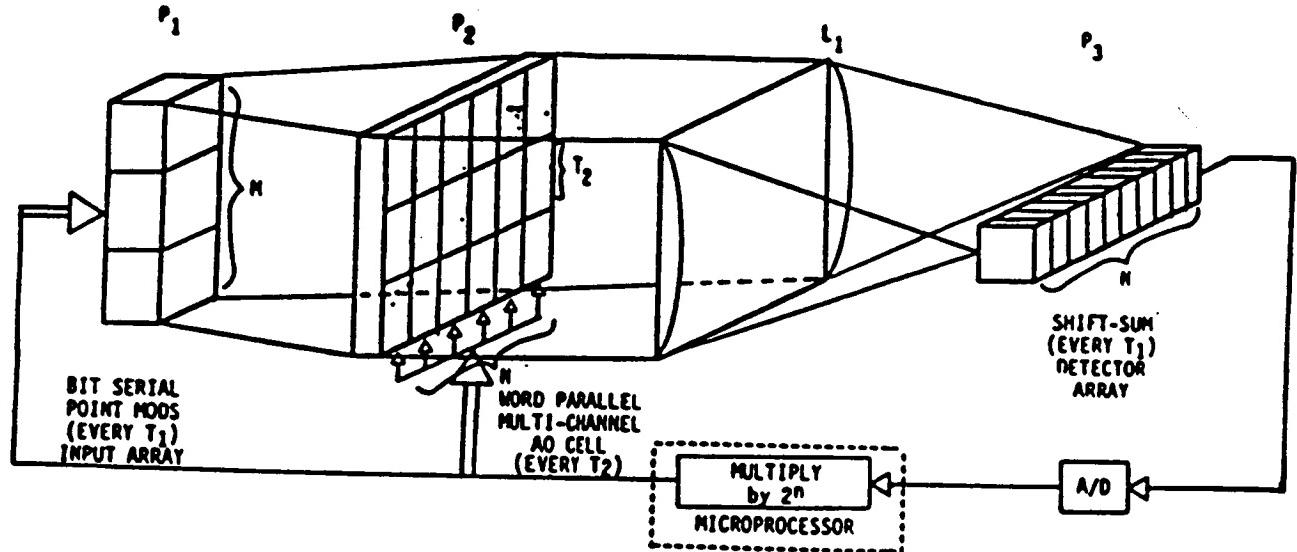
## 5.2 Architecture Review

The optical laboratory matrix-vector array processor emphasizes the well-known digital multiplication by analog convolution (DMAC) algorithm<sup>9, 10</sup>. The system of Fig. 5-1 uses detector time integration to achieve this with one encoded data stream fed time-sequentially to one  $P_1$  point modulator and with the second number data representation fed in parallel to AO2 in  $P_2$ . The  $P_3$  detector system performs a shift and add electronically to produce the convolution of the two input data sequences by time integration and shifting on the detector. This is attractive, because one output bit of the final mixed-radix product number is produced each bit time  $T_B = T_1$  (the time each  $P_1$  modulator is pulsed on). This architecture is attractive for four major reasons:

1. Both space and time integration are utilized with multi-channel processing.
2. Only one out A/D and adder are required and data flow is ideal.
3. Various number representations are possible on the same architecture.
4. A novel LU decomposition algorithm can be realized on the same optical system.

We denote the number of  $P_1$  channels by  $M$  and the number of  $P_2$  input channels by  $N$ . Each of the  $M$  channels produces one high-accuracy product and all  $M$  channels produce a high-accuracy vector inner product (VIP) (with one output bit each  $T_B$ ). Operation of the system is as follows. Each  $P_1$  output uniformly illuminates one row of  $P_2$  data. The convolution of the  $P_1$  and  $P_2$  data appears time-sequentially at the single-channel  $P_3$  output. This process occurs in parallel for all  $M$  input channels to yield an  $M$  element VIP output. The

DMAC encoding produces this output to high accuracy. New bit data is fed bit serial to AO1 at  $P_1$ , each  $T_1$  and after  $NT_1$  of time, one word or encoded number is produced. New data is fed bit-parallel to AO2 at  $P_2$  each  $T_2 = NT_1$  and a VIP output results each  $T_2$ , with one bit of the VIP produced at  $P_3$  each  $T_1 = T_2/N$ . A full VIP is produced each  $T_2$ . In practice, we feed the second number to  $P_2$  twice each  $T_2$  as pulses  $T_1$  long each  $T_2/2$  of time. For the general laboratory system, we employ an  $M=10$  channel AO cell at  $P_1$  and an  $N=32$  channel AO cell at  $P_2$ . For the initial laboratory design,  $M=10$  and  $N=32$ . Both AO cells exist in our laboratories (AO1 is a point modulator with 100MHz bandwidth and AO2 has an aperture  $T_A=5\mu s$  and a bandwidth of 200 MHz). Both cells have center frequencies of about 300 MHz.



**Figure 5-1:** Multi-Channel High Accuracy Time and Space Integrating Architecture<sup>6</sup>

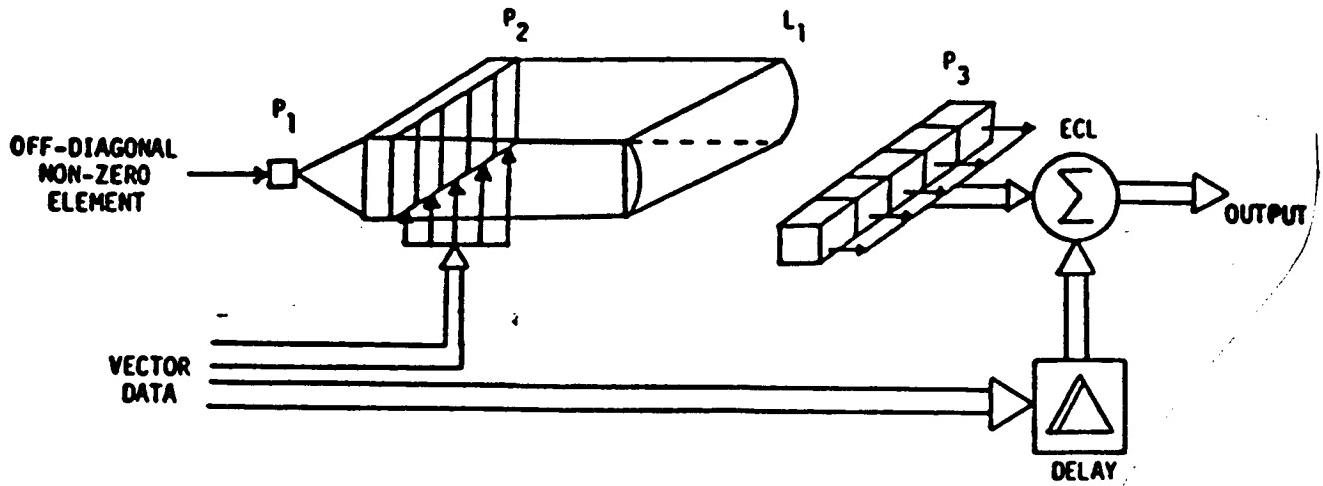
### 5.3 Number Representation

The DMAC algorithm can be applied to any base<sup>6</sup> and such techniques represent considerable speed improvements as well as hardware reductions<sup>6</sup>. For example, with  $N=10$  channels at  $P_2$  and  $L=2$  levels (Base  $B=2$ ), we achieve  $L^N=2^{10}=10$  bits of accuracy. However, with  $N=10$  and  $B=3$  ( $L=8$ ), we can realize  $L^N=8^{10}=30$  bits or higher accuracy with no

reduction in speed or performance. If we reduce N to N=6 and still employ B=3 (L=8), we can achieve  $8^6=18$  bits of accuracy with fewer channels (N=6). This reduces hardware as well as speeding up the multiplication throughput (VIP multiplication time  $T_2=NT_1$  is also reduced since N is reduced). The hardware requirements are also reduced (since N=6 versus 10 channels of  $P_2$  input RF electronics and N=6 versus 10 channels of output  $P_3$  detector channels are now required). This is achieved with an increase in the A/D  $P_3$  requirements. Floating point accuracy<sup>23</sup> is achieved by optically processing the mantissas and with digital processing performed on the exponents.

Our present case study and application requires only bipolar data. This can be achieved by use of sign-magnitude<sup>6</sup>, 2's complement<sup>1</sup>, negative base<sup>4</sup> or by use of biased<sup>24</sup> data representations. As detailed in the references noted, different number representations are more suitable for parallel channel architectures in which data is summed on the output detector, than are other number representations. In the specific case study of concern here, we consider a direct LU matrix decomposition algorithm<sup>6</sup> which requires only one channel of the system of Fig. 5-1 [see Fig. 5-2 and reference 6]. The system of Fig. 5-1 is attractive since it allows various encoding schemes to be implemented on the same architecture. We employ different techniques in different stages and for different uses of the system. We use sign-magnitude number representation for the LU decomposition algorithm we emphasize here. The other number representations follow directly and can be used as necessary for a given application. For complex-value data, the three-tuple or four-tuple number representation is used as required. We note (as emphasized in other publications<sup>1, 4, 24</sup>) that multi-channel processors such as that of Fig. 5-1 are necessary to achieve the higher number of multiplications per second needed for optical systems to compete with digital parallel and multiple processor systems.

For the present system, AO2 has an aperture time  $T_A=5\mu s$ , which we divide into M



**Figure 5-2:** One-Channel LU Decomposition Architecture for Matrix Decomposition<sup>6</sup>

regions, i.e.  $MT_2=10T_2 \leq 5\mu s$ , i.e.  $T_2=0.5\mu s$ . In our general laboratory system, we use  $T_2=0.25\mu s$  (a 4 MHz data rate) and feed repeated data (each pulse  $T_1$  long) to AO2 each  $T_2/2$  and we feed new bit data to each  $P_1$  channel each  $T_1=T_2/M=0.025\mu s$ . The use of two  $P_2$  pulses in each  $T_2$  insures that one AO2 data packet is present during one  $T_2/2$  interval. The TeO<sub>2</sub> longitudinal AO cells used in  $P_1$  and  $P_2$  satisfy the above requirements. Thus, the system design allows input  $P_1$  data to AO2 each  $T_1=25ns$  (at 40 MHz) and new  $P_2$  data to AO2 each  $T_2=MT_1=250ns$ .

## 5.4 Partitioning, LU Decomposition and Accuracy Tradeoffs

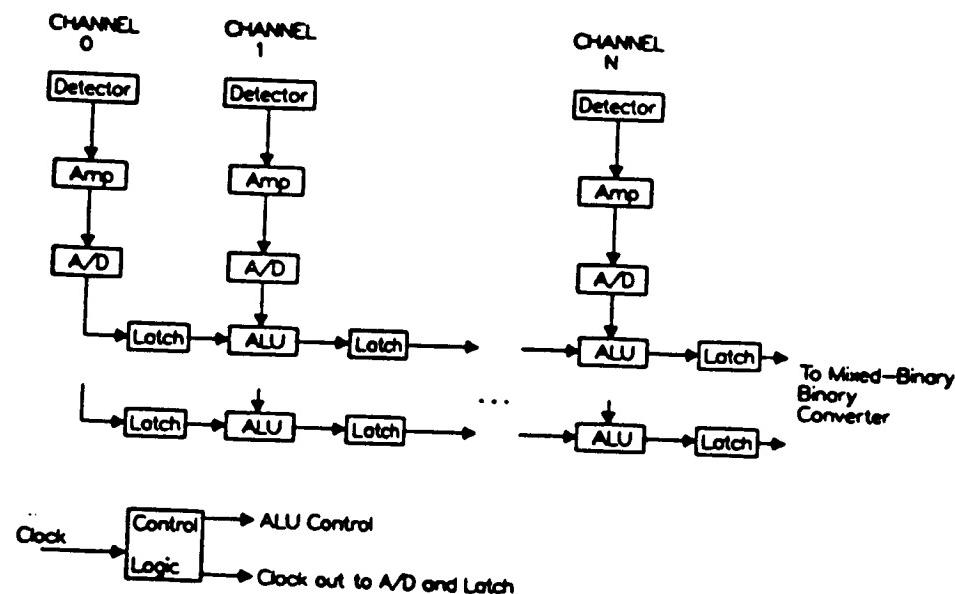
### 5.4.1 Diagonal Partitioning

To allow partitioning of matrices whose size exceeds the number of input  $P_1$  elements in AO1, we partition the matrix along its diagonals<sup>6</sup>. In other cases, we feed the proper matrix data to the  $P_1$  point modulators each  $T_B$  as detailed elsewhere<sup>13, 21</sup>. We have also detailed a multi-processor architecture suitable for partitioning of matrices with multiple banded structure<sup>21</sup>.

### 5.4.2 Output P<sub>3</sub> Flexible Detector System

The P<sub>3</sub> detector system used is shown in Fig. 5-3. It employs a separate A/D converter, latch and ALU on each detector. This system allows one to form the shift/add that produces the mixed radix output. With this output system, each output digit is now binary encoded. This simplifies conversion to conventional binary. This output system is quite flexible. It also allows use of a novel 2's complement negative number representation<sup>1</sup>, negative base<sup>4</sup>, the ability to dump the output detector contents in parallel to avoid output P<sub>3</sub> dead time (this is achieved by the second latch/ALU system shown in Fig. 5-3).

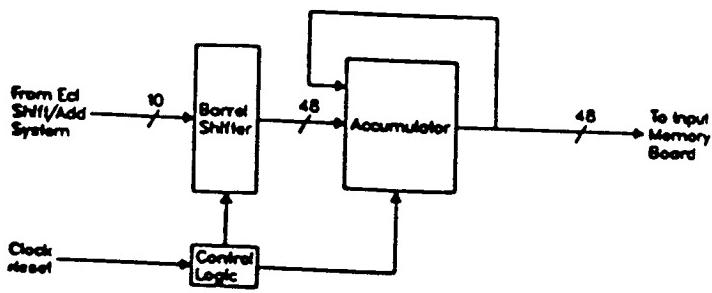
This detector system also simulates a high-speed GaAs CCD detector system and other related output architectures, with the ability to change detectors and other components with considerable flexibility.



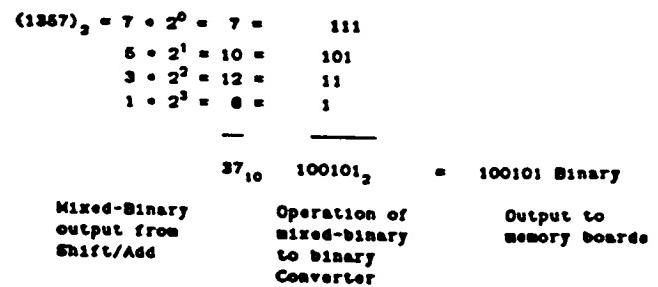
**Figure 5-3:** Output P<sub>3</sub> Detector System with Number Representation and Component Flexibility and with a New Conversion Algorithm Ability

With the output mixed-radix data from the P<sub>3</sub> system of Fig. 5-3 available as separate 10-bit digits (available in parallel; each is the 10-bit digitally-encoded version of one of the N mixed-radix output digits), a simple conversion to conventional binary results. The required

circuitry (Fig. 5-4)) is quite simple. The algorithm required is also simple: for the first word (the binary representation of the least significant digit) we perform no shift and merely input this digital data to the accumulator at time  $T_1$ ; for the second word, we shift the output data by one bit (this is achieved in the parallel barrel shifter) and add this shifted word to the accumulator at bit time  $2T_1$ ; the third word is shifted by two bits (in the barrel shifter) and added to the accumulator contents; etc. Figure 5-5 shows an example of the operations required and performed (the shift and add of successive output data and the output of one bit of the result per  $T_1$ ).



**Figure 5-4:** Schematic of a Simplified Output Binary Conversion Hardware System.



**Figure 5-5:** Example of Output data conversion in  $P_3$  Output System.

#### 5.4.3 LU One-Channel Algorithm<sup>6</sup>

To solve  $\underline{A} \underline{x} = \underline{b}$  for  $\underline{x}$  by LU decomposition, we decompose the matrix  $A$  into  $\underline{A} = \underline{L}\underline{U}$  (where  $\underline{L}$  and  $\underline{U}$  are lower and upper triangular matrices). This allows us to solve the original problem by back substitution. The decomposition is achieved by multiplying  $\underline{A}$  by  $N$  decomposition matrices  $P_m$  (when  $\underline{A}$  is  $N \times N$ ). Synthesis of the decomposition matrix is trivial and requires only calculation of one column of the prior  $P_m \underline{A}_{m-1}$  matrix (where  $\underline{A}_{m-1} =$

$P_{m-1}A_{m-2}$ ). The realization of the associated matrix-matrix multiplication  $P_m A_{m-1}$  required by the algorithm can be achieved on the one channel system of Fig 5-2 (one channel of the architecture of Fig 5-1). We consider implementation of this algorithm with an augmented matrix (with a matrix  $A$  augmented with  $b$  as one column). This produces: one row of the matrix  $U$  and one element of the new  $Ux = b'$  vector each  $T_2$ . These outputs can feed a separate lower triangular (back substitution) processor as shown in Fig. 5-6. At each  $T_2$ , the optical system of Figure 5-2 outputs a new row of the next  $A_m$  matrix required to calculate one element of the unique column of the next decomposition matrix  $P_m$  as shown in Fig. 5-7. The calculation of the necessary one column of  $P_m$  is straight-forward and is detailed elsewhere<sup>6</sup>.

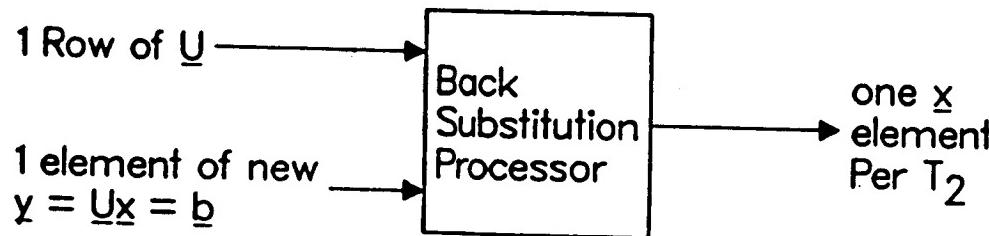


Figure 5-6: Post Processing (for Back-Substitution Solution) Required Per  $T_2$

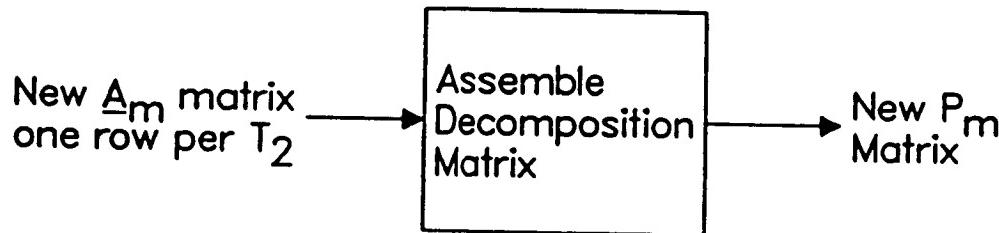


Figure 5-7: Post Processing (to Compute New Decomposition Matrix  $P_m$ ) each  $T_2$

#### 5.4.4 Accuracy Above the Number of Channels by Partitioning

To achieve greater than  $2^N$  bit precision on a  $N$  bit DMAC processor operating in base 2, we proceed as follows. We convolve the first  $N$  bits of the two numbers and store the results. We then input the next  $N$  bits of the two numbers, convolve the results and accumulate these

and the prior N convolved bits. This procedure repeats for the number of cycles needed. By this technique (utilized in our laboratory system), we achieve an accuracy above the number of bits available in the processor. This allows us the added flexibility of a different number of system bits and final system accuracy (thus allowing a hardware/accuracy and speed trade-off). For the system fabricated and the example performed, we use an  $N=3$  bit system to produce  $B=21$  bit accuracy. This requires  $(B+N-1) \cdot (B/N)T_1 = (21+3-1)7T_1 = (23T_1)7 = 7T_2$  of time. This fully utilizes the available capacity of the processor. This is possible and is a unique feature of the DMAc algorithm (since carries in it do not occur and need not be handled until in the final mixed radix to binary conversion).

## 5.5 General Laboratory Electronic Support System Requirements

The laboratory electronic support system fabricated used an Intel 286/380 system. This system runs the iRMX86 operating system with support for C, Assembler, PLM and Fortran. The hardware includes a VAX interface to download data to the processor. Hardware and support boards and equipment include: an Intel 286/10 single board processor with a 80286 16-bit microprocessor, an Intel 80287-8 mathematical co-processor for floating point and trigonometric calculations, one M-byte of one-wait-state RAM memory, a 2-port serial card, an intelligent disk control card, a tape control card, two cards for memory subsystem control, two 32 M-byte hard disks, a 1.2 M-byte floppy disk, an 0.5 inch tape drive and output display facilities.

The electronic hardware concept used employs burst processing, in which input data will be fed to the optical processor at the memory data rates through multiplexors at high data rates (40 MHz) through a 4 bit 200 MHz D/A for multi-level  $P_1$  and  $P_2$  input data. The data for  $P_1$  and  $P_2$  will be provided from parallel buffer memories for  $P_1$  and  $P_2$  input data. These data will feed the optical processor. The  $P_3$  output data is collected by 6 bit 100 MHz A/D's through the

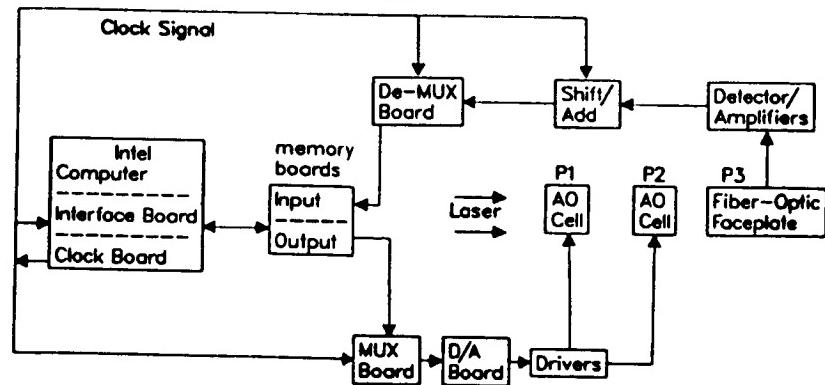
shift/add array to input buffer memory channels. These buffer memory channels are provided on other memory boards. Each input and output memory board provides eight channels of 12 bit data at 10 MHz ( $0.1 \mu s$ ) per channel for  $96 \times 10 = 960M$  bits per second per card. With the eight cards available, we can achieve approximately 8G bit per second data rate generation. Figure 5-8 shows a block diagram of the full processor and Fig. 5-9 shows a photograph of the hardware support. Figure 5-10 shows the  $P_3$  hardware in Fig. 5-3. Figure 5-11 shows a close-up of the  $P_3$  hardware board.

## 5.6 Electro-Optical Laboratory System

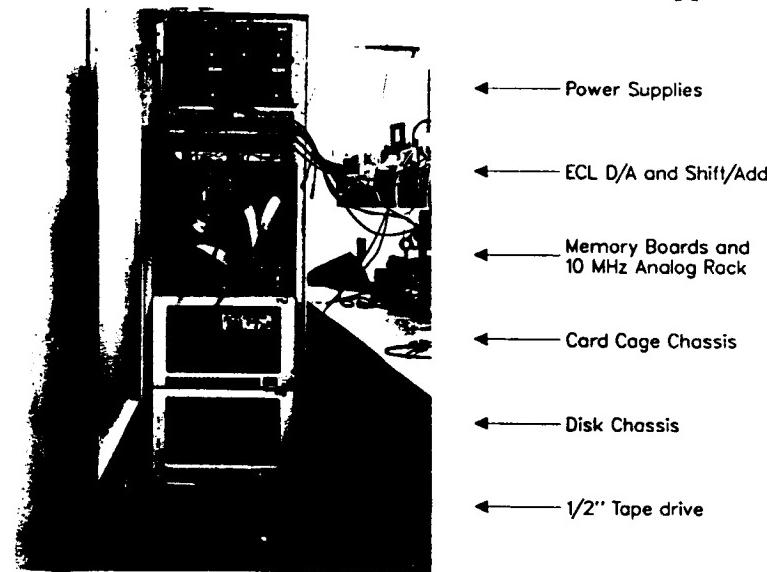
The optical laboratory system used in the tests reported is shown in Fig. 5-12. Only  $M=1$  channel of the 10 channel AO1 cell at  $P_1$  was used (since our LU algorithm requires only a single-processor channel system). Only  $N=3$  channels of the 32 channel AO2 cell at  $P_2$  were used (to demonstrate the ability of this system to achieve 20 bit accuracy on a 3 bit channel system using our partitioning algorithm). The laboratory system was operated with  $P_1=0.1 \mu s$  and  $T_2=(N+B-1)T_1=23T_1$  to demonstrate high accuracy. For multi-level encoded data tests,  $L=3$  levels of the  $2^4$  levels possible from the input D/A were used. This allows the output D/A levels used to be adjusted for processor nonlinearities and noise. The output A/Ds were 6 bit 100 MHz units (one per detector as shown in Fig. 5-3).

## 5.7 Finite Element Case Study

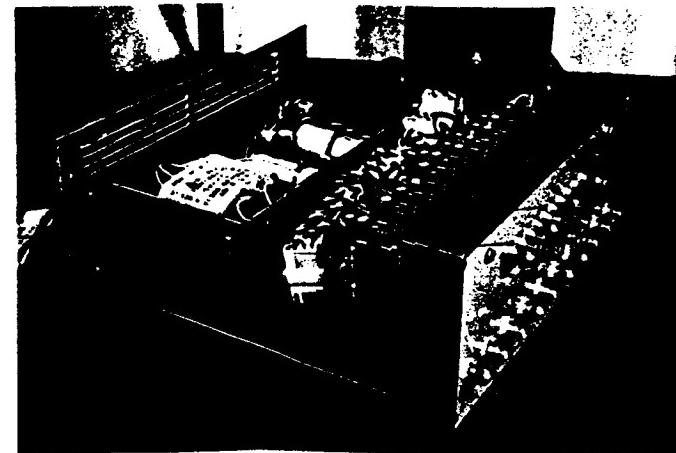
The finite element case study involved the solution of the system of LAEs  $\underline{K} \underline{d} = \underline{p}$  for  $\underline{d}$ . The problem chosen was detailed elsewhere<sup>25</sup>. Here,  $\underline{K}$  is the  $N \times N$  stiffness matrix that defines the structure of the system and the relationships between the finite elements that model the structure,  $\underline{p}$  is an  $N \times 1$  vector that defines the  $N$  possible loads or forces on the structure and  $\underline{d}$  is the desired output  $N \times 1$  vector of the displacements (3 per node) produced at the nodes of the structure described by  $\underline{K}$  with the forces described by  $\underline{p}$  applied. The problem considered was an aluminum plate  $6' \times 8' \times 1"$  divided into 8 rectangular plate bending finite element regions as



**Figure 5-8:** Block Diagram of the Electronic Support System

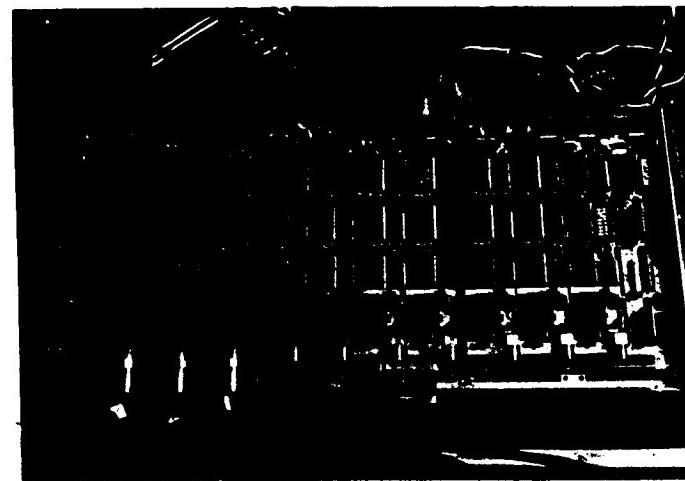


**Figure 5-9:** Photograph of the Electronic Support System



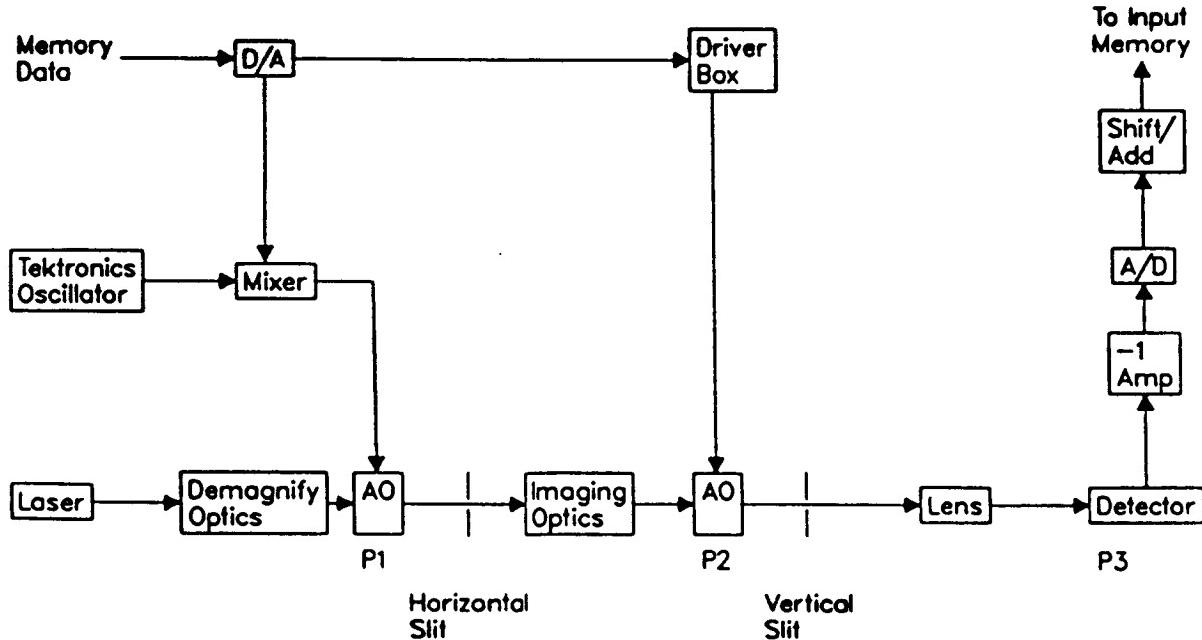
**Figure 5-10:** Photograph of P<sub>3</sub> Hardware

ORIGINAL PAGE IS  
OF POOR QUALITY



ORIGINAL PAGE IS  
OF POOR QUALITY

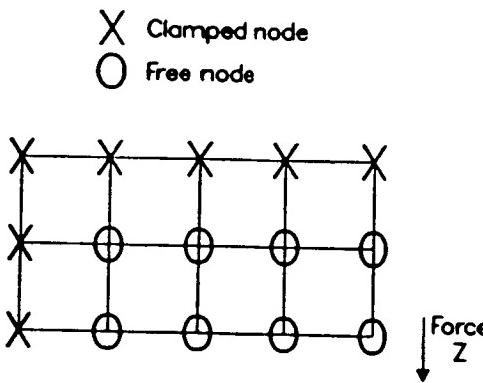
**Figure 5-11:** Photograph of  $P_3$  Hardware Board



**Figure 5-12:** Block Diagram of the Reduced Laboratory System Used in the Demonstrations Described

shown in Fig. 5-13. The structure has  $M=15$  nodes with  $D=3$  degrees of freedom (displacements, etc.) per node for a total of  $N = M \times D = 45$  degrees of freedom to be described for the system. The matrix  $K$  is  $45 \times 45$ . The matrix bandwidth is reduced to 29 by optimal node numbering. The boundary conditions used involved clamping two edges of the structure (the nodes denoted by  $x$  in Fig. 5-13) with a force applied in the  $z$  direction at the bottom right

node (case 1) and at this node and the adjacent edge nodes (case 2). The elements required in the  $\underline{d}$  solution vector were the 3 degrees of freedom at the 8 unclamped nodes (24 unknowns). From calculations, the dynamic range of  $\underline{K}$  was found to be  $10^5$  (17 bits). We estimated that 21 bits of precision were necessary to solve for  $\underline{d}$  to reasonable accuracy and to allow processing of  $\underline{K}$  to reasonable accuracy. We achieve 21 bit accuracy on our  $N=3$  bit channel system with a partitioning technique described earlier using  $(B+N-1)(B/N)T_1 = (21+3-1)(7T_1) = (23T_1)7 = 7T_2$  of time, where  $B=21$  is the number of bits desired and the  $N-1$  term accounts for the fact that the convolution of  $N$  bits is  $2N-1$  bits long. We use  $T_2 = 23T_1$  as noted before. Thus, by running the system 7 times ( $T_2$  of time for each run) with 3 bits produced per  $T_2$ , we achieve 21 bit final accuracy.



**Figure 5-13:** The Aluminum Plate Finite Element Structure Used for Our Case Study

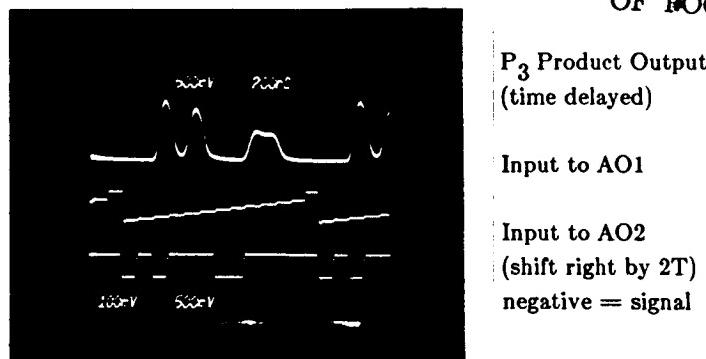
## 5.8 Laboratory System Data

The laboratory data of Fig. 5-14 shows the system's ability to accurately process multi-level data. Trace 3 shows the binary input sequence to AO2 (negative pulses indicate the presence of a 1). Trace 2 shows the multi-level time sequential input signal to one channel of AO1 (more negative values on the scope trace are more positive numbers). Trace 3 should be shifted right by two time slots (the delay required for this data to reach the time aperture region of AO2 illuminated by the AO1 channel used) to time align the plots. The top trace 1 shows the  $P_3$  output obtained. It is the expected product of the AO2 and AO1 data, i.e. the AO1 data in time slots 3, 5, 9 and 10 (the times when the AO2 data is 1 and not 0).

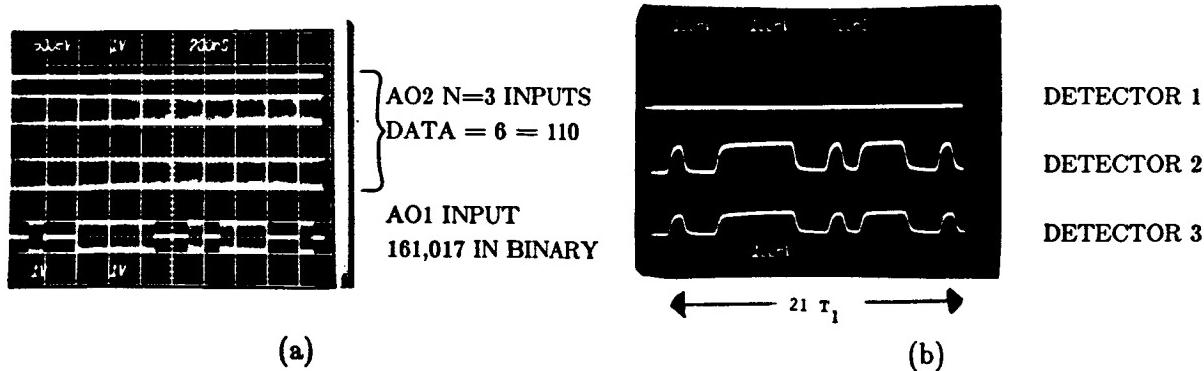
Figure 5-15 shows the system's ability to convolve two digital bit streams and hence DMAC processing of high accuracy data. The AO1 time-sequential input to one channel was 161,017 or in binary the 21 bit sequence 000100111010011111001. The AO2 input number was 6 or in binary 110. The AO1 input is shown in trace 4 of Fig. 5-15a. The inputs to the three channels of AO2 are shown in traces 1 to 3 (they are 011 respectively or the binary version of the second number to be processed). The output time history on the three detectors at  $P_3$  (opposite the corresponding regions of AO2) are the products of the AO1 time sequence and the AO2 bits (1 or 0). These results (Fig. 5-15b) are 0 (for detector 1 opposite the AO2 channel with input 0) and the input 21 bit sequence (for the other 2 detectors).

Figure 5-16 shows the DMAC algorithm example performed on the laboratory system. The mixed radix output data from the laboratory system is shown in Fig. 5-17a. The digital representation of each mixed radix digit is produced in the  $P_3$  system of Fig. 5-3 and is shown in the A0 and A1 output waveforms which represent the two binary digit versions of the mixed radix output (bit-by-bit). The final conventional binary representation of the output obtained on the laboratory system is shown in Fig. 5-17b.

The results for our two finite element case studies are shown in Figs. 5-18a and 5-18b respectively. Column 1 lists each of the 24 internal node degrees of freedom to be calculated. The values calculated to floating point accuracy on a VAX using IMSL algorithms (column 2) and on our Intel simulator (column 3). This verifies the accuracy of our Intel processor algorithm. The results calculated to 21 bit accuracy on the digital system (column 4) and on the optical laboratory system (column 5) also agree. This verifies the intended 21 bit accuracy of our optical processor in a full engineering problem.



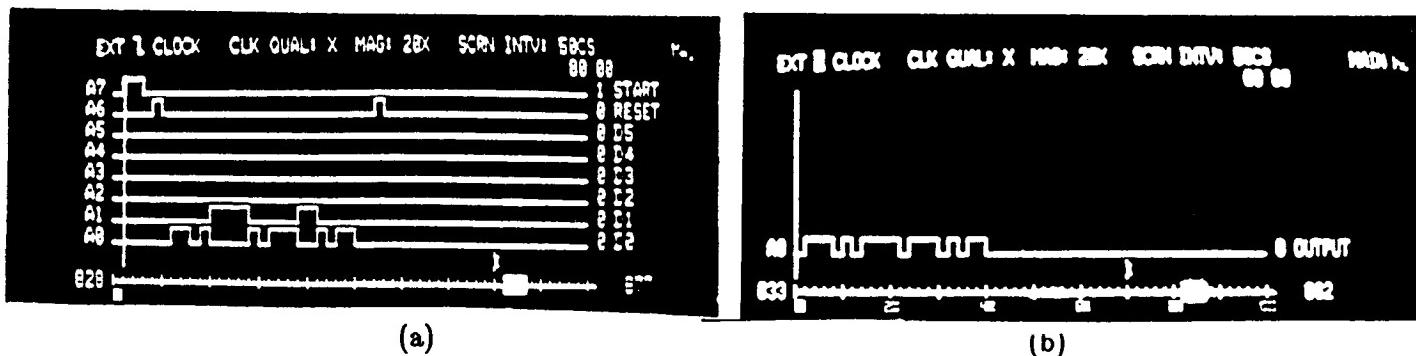
**Figure 5-14:** Demonstration of Multi-Level Data Handling and Multiplication on the Laboratory System



**Figure 5-15:** Input (a) and Output (b) Data for High-Accuracy DMAC on the Laboratory System

000100111010011111001	A01 input
110	A02 inputs
<hr/>	
000000000000000000000000	Detector 1 output
000100111010011111001	Detector 2 output
000100111010011111001	Detector 3 output
<hr/>	
0001101221101222210110	Shift/Add output
<hr/>	
00011101011110111010110	Final binary output

**Figure 5-16:** Representative DMAC Example Performed on the Laboratory System



**Figure 5-17:** Binary Representation (a) of the Mixed Radix Output (only Channels A0 and A1 are active for this example) and (b) Conventional Binary Decoded Version of the Output

	Solution Vector	MSL Results Fl. point	Intel Simulator Results Fl. point	Intel Simulator Results 21 bits	Optical Results 21 bits
Vector 1	x(0)	26.439	26.439	25.139	25.139
	x(1)	-0.328	-0.328	-0.318	-0.318
	x(2)	0.478	0.478	0.460	0.460
	x(3)	9.923	9.923	9.324	9.324
	x(4)	-0.133	-0.133	-0.123	-0.123
	x(5)	0.422	0.422	0.400	0.400
	x(6)	18.467	18.467	17.459	17.459
	x(7)	-0.349	-0.349	-0.335	-0.335
	x(8)	0.357	0.357	0.340	0.340
	x(9)	6.382	6.382	5.967	5.967
	x(10)	-0.147	-0.147	-0.138	-0.138
	x(11)	0.297	0.297	0.280	0.280
	x(12)	9.959	9.959	9.334	9.334
	x(13)	-0.337	-0.337	-0.318	-0.318
	x(14)	0.193	0.193	0.181	0.181
	x(15)	3.201	3.201	2.974	2.974
	x(16)	-0.118	-0.118	-0.110	-0.110
	x(17)	0.160	0.160	0.150	0.150
	x(18)	3.048	3.048	2.834	2.834
	x(19)	-0.225	-0.225	-0.210	-0.210
	x(20)	0.043	0.043	0.040	0.040
	x(21)	0.877	0.877	0.809	0.809
	x(22)	-0.071	-0.071	-0.066	-0.066
	x(23)	0.053	0.053	0.049	0.049

	Solution Vector	MSL Results Fl. point	Intel Simulator Results Fl. point	Intel Simulator Results 21 bits	Optical Results 21 bits
Vector 2	y(0)	13.437	13.437	12.767	12.767
	y(1)	-0.209	-0.209	-0.204	-0.204
	y(2)	0.265	0.265	0.255	0.255
	y(3)	4.443	4.443	4.140	4.140
	y(4)	-0.055	-0.055	-0.051	-0.051
	y(5)	0.210	0.210	0.199	0.199
	y(6)	8.559	8.559	8.038	8.038
	y(7)	-0.193	-0.193	-0.186	-0.186
	y(8)	0.167	0.167	0.158	0.158
	y(9)	2.857	2.857	2.646	2.646
	y(10)	-0.068	-0.068	-0.064	-0.064
	y(11)	0.137	0.137	0.128	0.128
	y(12)	4.343	4.343	4.020	4.020
	y(13)	-0.155	-0.155	-0.145	-0.145
	y(14)	0.062	0.062	0.076	0.076
	y(15)	1.380	1.380	1.264	1.264
	y(16)	-0.053	-0.053	-0.049	-0.049
	y(17)	0.070	0.070	0.065	0.065
	y(18)	1.287	1.287	1.176	1.176
	y(19)	-0.096	-0.096	-0.089	-0.089
	y(20)	0.018	0.018	0.016	0.016
	y(21)	0.363	0.363	0.328	0.328
	y(22)	-0.030	-0.030	-0.027	-0.027
	y(23)	0.022	0.022	0.020	0.020

(a)

(b)

Figure 5-18: Case Study 1 (a) and Case Study 2 (b) Data Digitally and Optically Calculated

## 5.9 Summary and Conclusion

The lengthy experiments and data described have provided many new results. A new optical architecture has been fabricated. It allows the use of multi-level and binary DMAC (both were experimentally demonstrated). Its output  $P_3$  detector system produces binary-encoded versions of each mixed-radix output digit. This allows easy conversion to the final binary form (this was also experimentally demonstrated). The DMAC algorithm and this output format allows a new partitioning technique to increase accuracy without increasing the number of digit channels required. This is possible since the DMAC algorithm carries need not be performed until the final binary conversion is implemented. We demonstrated in the laboratory the calculation of 21 bit accurate data with a 3 bit (digital channel) system using this tradeoff of speed, hardware and accuracy. The resultant architecture allows a variable accuracy processor that can accommodate many new algorithms and number representations. We performed a

laboratory demonstration using a new one-channel LU decomposition algorithm (which, using an augmented matrix, provides output data for the final back substitution step with perfect data flow). We produced the first laboratory processing of a finite element problem, the first laboratory direct LAE solution, and the first use of a multi-channel AO laboratory matrix-vector system.

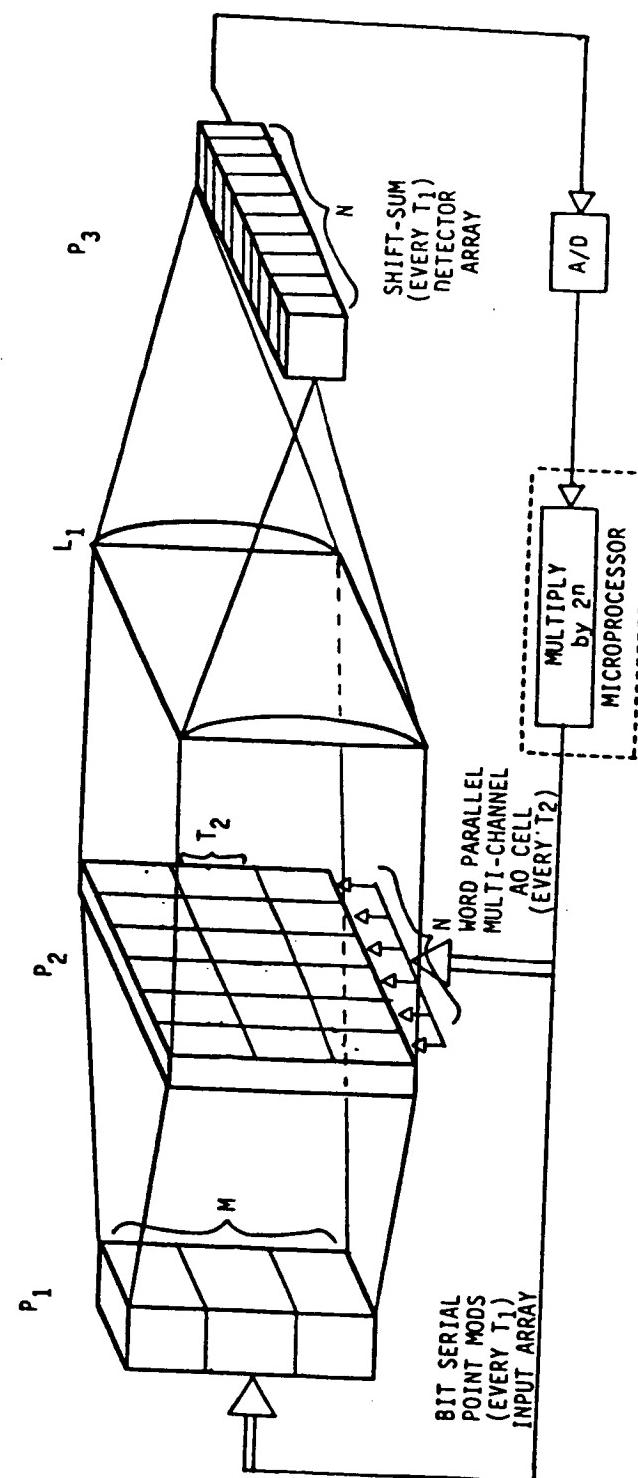
## 6. Multi-channel Encoded System Design and Fabrication

In this chapter we describe the optical system and the software support for it. We first describe the proposed optical architecture and the electronic support requirements. Then we describe the special hardware constructed to run the system. Next the software support is considered and we explain how the software was made more user-friendly than previous systems. Lastly, the actual system built is described including results that were obtained.

### 6.1 Architecture

The optical system in Figure 6-1<sup>6</sup> consists of a linear array of M point modulators at  $P_1$ . These are imaged vertically and expanded horizontally onto  $P_2$ , which contains an N element AO cell. For discussion purposes, we consider M vertical regions of the AO cell at  $P_2$ . Each  $P_1$  point modulator uniformly illuminates one horizontal region of all N AO channels at  $P_2$ . Plane  $P_2$  is imaged horizontally and integrated vertically onto  $P_3$ . For simplicity, we consider  $P_3$  to contain a shift register linear detector array. The exact  $P_3$  system is detailed in Section 6.2.3.

To achieve the accurate product of two binary-encoded numbers on one channel ( $M=1$ ) of the system in Figure 6-1, the bits of one number  $s_2$  are fed word parallel to  $P_2$  and the bits of the other number  $s_1$  are fed serially to  $P_1$ . The  $P_3$  output is the convolution of  $s_2$  and  $s_1$ . For N-bit words, a new bit enters  $P_1$  each  $T_1$  of time and one word is entered at  $P_2$  each  $NT_1=T_2$ . The 1-D data incident on  $P_3$  each  $T_1$  is  $s_2$  or zero (depending on the input at  $P_1$ ). Each  $T_1$ , the contents of  $P_3$  are shifted by one location and the new 1-D  $s_2$  data at the present  $T_1$  are added to the shifted outputs of the prior  $T_1$ . This system thus achieves the summation of all proper partial products in the multiplication of two encoded numbers by shifting and time integration on the detector. One new mixed-binary output digit is produced each  $T_1$  and the full output is available after  $NT_1$ . A CCD (charge couple device) shift register detector and one output A/D



**Figure 6-1:** Multi-Channel AO Cell Architecture

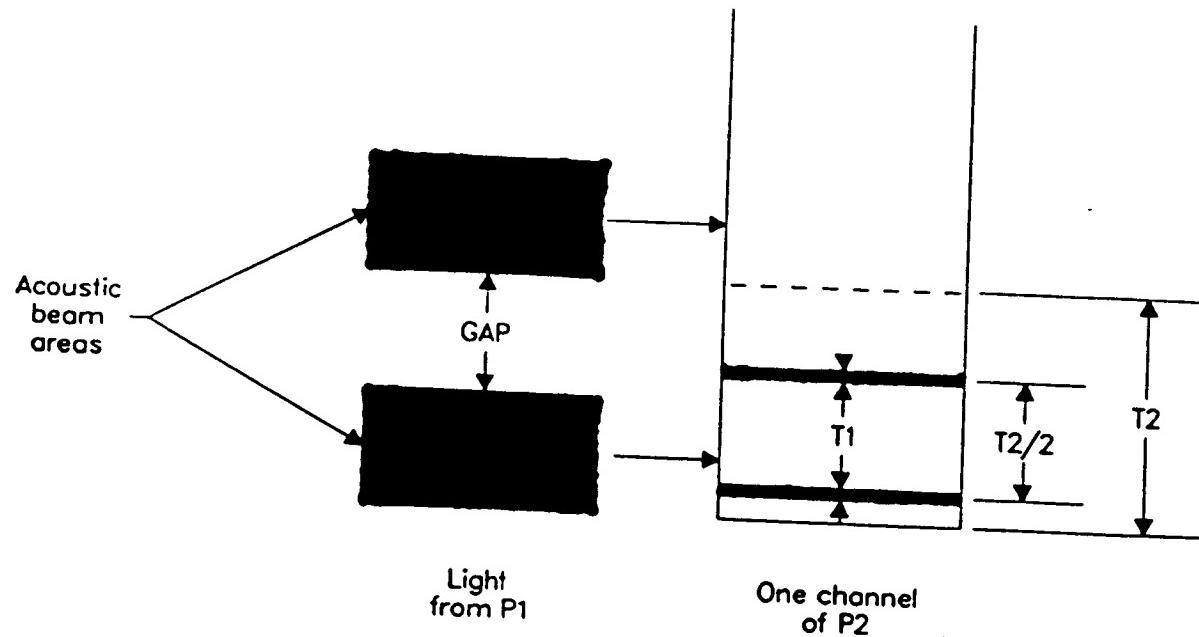
converter can achieve the required detection function. We consider the design and electronic support requirements for a system with  $M=10$  and  $N=10$  in Section 6.2.

## 6.2 Electronic Support Requirements for Multi-Channel Encoded System

We now detail the electronic support requirements and capability of the multi-channel encoded system. For each  $P_1$  and  $P_2$  channel we allow up to three bits ( $L = 8$  levels) of input data. We allow up to  $M=10$  channels at  $P_1$  and  $N=10$  channels at  $P_2$ . For multi-channel algorithms ( $M>1$ ), we use only  $L=3$  input levels, since the output A/D on each detector is 6 bits. Section 6.2.3 details these calculations. In this section, we describe the full  $M=10$  and  $N=10$  system. This allows us to specify the full size and capabilities of the support hardware.

### 6.2.1 AO Cells and A/D Converters

The electronic support system was designed to support 10 channels of a 32 channel AO cell at  $P_1$  with a center frequency  $f_c = 300$  MHz and a bandwidth of 100 MHz. This cell is a  $\text{TeO}_2$ , longitudinal mode device used as a point modulator. The  $P_2$  cell is also a  $\text{TeO}_2$  longitudinal mode device with a center frequency  $f_c = 400$  MHz, bandwidth  $BW_A = 200$  MHz and length  $T_A = 5\mu\text{s}$ . Again only ten channels at  $P_2$  are considered here. With  $M=10$  and  $T_A = 5\mu\text{s}$ ,  $MT_2 \leq T_A$  and  $T_2 \leq 0.5\mu\text{s}$ . We also require  $T_2 = NT_1 \leq 0.5\mu\text{s}$  or  $NT_1 = 10T_1 \leq 0.5\mu\text{s}$  or  $T_1 \leq 0.05\mu\text{s}$ . The  $P_3$  design for  $T_1 = 0.05\mu\text{s}$  is possible. We are presently using  $T_1 = 0.1\mu\text{s}$  because the high speed ALU chips necessary for the  $P_3$  detector were not available, nor were equipment funds. With  $T_1 = 0.1\mu\text{s}$ , we are limited to  $M=5$  channels in  $P_1$  and thus  $M=5$  regions in  $P_2$ . The  $P_2$  channels are fed with data each  $T_2$ . There is a gap between channels on the  $P_1$  AO cell that is also equal to the width of each of the acoustic columns in  $P_1$  as shown in Figure 6-2. Because of this, we feed data to  $P_2$  as a pulse of duration  $T_1$  every  $T_2/2$  (i.e. 2 pulses  $T_2/2$  apart each  $T_2$ ). This insures that one data pulse is always present in  $P_2$  opposite an active region of  $P_1$ .



**Figure 6-2:** Diagram of acoustic gaps from  $P_1$  and data inputs to  $P_2$

The number of levels  $L$  used per digit and the number of digits  $N$  used in the encoding determines the system accuracy,  $L^N$ . With  $L=8$  (3 bits) and  $N=10$ , we have a dynamic range or accuracy of 1,073,741,824 (30 bits). With  $L=3$ , the accuracy is reduced to 59049 (15 bits). It is possible to double the dynamic range by simply running the data through the system twice. In the first cycle the  $N$  least significant digits are processed and in the second cycle the  $N$  most significant digits are processed. This method is also used in our laboratory demonstration system (Section 6.10). This would increase the accuracy on an  $N=10$  channel  $P_2$  system to 3,486,784,401 (greater than 31 bits). This is expected to be adequate for all initial applications to be considered. The number of channels  $M$  together with  $L$  and  $N$  determine the A/D requirements at  $P_3$ . For the system designed we use an input D/A with 4 bits at 200 MHz. The data rate satisfies our  $T_1$  data design (25 ns or 40 MHz). For the system design, we use a 6 bit 100 MHz A/D in  $P_3$ . This is fast enough for the 40 MHz  $P_3$  shift rate. With this A/D, we can

allow input values of  $L=8$  for  $M=1$  channels at  $P_1$  or  $L=3$  for  $M=10$  channels. Section 6.2.3 details the  $P_3$  requirements.

### 6.2.2 Input Data Requirements

For  $M=N=10$  and  $T_A=5 \mu s$  for the AO cell at  $P_2$ , we found  $T_1 \leq 50 \text{ ns}$  was required. We designed the input electronics to allow a  $T_1=25 \text{ ns}$  (40 MHz) input data rate and three bits (eight levels) per  $T_1$  digit. The  $T_1$  rate used in the initial lab system is slower, but this section considers what the system can provide. This is equivalent to a  $40 \times 3 = 120 \text{ Mbit/sec}$  input data rate per channel. To achieve this, we plan to multiplex one 12-bit 10 MHz parallel output channel from our memory boards since its  $12 \times 10 = 120 \text{ Mbit/sec}$  data rate is exactly what is required for one  $P_1$  input at 40 MHz and 3 bits. With  $M=10$  channels at  $P_1$ , we require ten 12-bit 10 MHz buffer memory channels to provide all of the  $P_1$  data. The  $P_2$  input data rate is one-fifth that of  $P_1$  (one  $T_1$  pulse every  $T_2/2=5T_1$ ). We could thus use only two 12-bit 10 MHz memory channels for all  $N=10$  channels of  $P_2$  data. However, we use 10 channels because it simplifies timing. With 8 memory channels per memory board, we employ two memory boards for  $P_1$  and two for  $P_2$  with a multiplexer planned for each pair of memory boards.

### 6.2.3 Detector $P_3$ Requirements (future and now)

In the final embodiment of the system in Figure 6-1, the  $P_3$  detector system could utilize an advanced GaAs or VHSIC CCD shift register detector and partitioned A/D converters and adders. For the near term and in the laboratory system, separate detectors, A/D converters and adders are employed. If a system with  $N=M=10$  and  $L=8$  were used with one A/D, the A/D must resolve  $7 \times 7 \times 10 \times 10 = 4900 = 13$  bits. For a system with  $L=8$  (3 bits) and each  $P_3$  detector is sampled every  $T_1$ , each detector must resolve  $7 \times 7 \times 10 = 490$  levels. This case arises if all  $M=10$   $P_1$  inputs and all of the corresponding  $P_2$  data are the maximum value of 7. To resolve 490 levels would require a 9-bit A/D per detector. We are using a six-bit A/D on each detector and thus can detect 64 levels. Various new A/D converters are becoming available with

increased resolution and the required speed. We could also have multiplexed two detectors onto one A/D converter since they have more than double the needed speed. Since we have six-bit A/Ds, the system is limited to use of radix 3 numbers, ( $L=3$ ) when using  $M=10$  channels at  $P_1$ . For this case, the maximum output level possible is  $2 \times 2 \times 10 = 40$  which is less than 64. We have the capability to feed each of the M input  $P_1$  channels and each of the N input  $P_2$  channels of the optical system with three bits of data. This allows us to select three of the eight levels possible from the 3 bit  $P_1$  and  $P_2$  input data. This allows calibration routines to pick the three levels that give the most reliable output for the particular multiply since the optical and RF hardware has nonlinearities. For example, if the system was completely linear, three equally spaced numbers (0,4,7) would be picked out of the eight possible numbers. The system may work better though if numbers such as (0,2,7) were used instead.

The back end hardware fabricated consists of an amplifier, flash A/D converter, ECL adder (ALU) and latch per detector. Light is first detected and amplified to a suitable level for the A/D converters. The clock on the A/D converters is timed precisely with the input data to convert the detector's output at the proper time. The A/D output is then added (by the ALU) to the prior output (in the latch) of the stage immediately preceding it. At the first  $T_1$ , a reset operation is performed on the ALU, allowing the A/D data to pass directly into the latch with no add performed. At the next  $2T_1$ , the new  $P_3$  data is added by the ALU to the prior ( $1T_1$ ) data and the result is placed in the latches on the output of the ALU. All A/D converters are six bits wide, the first four ALU/latch combinations are eight bits wide and the last five ALU/latch sets are 12 bits wide. The output of the first A/D converter is fed directly into a latch since there is no previous data to add to it in this system. This is why there is one latch and 9 adder/latch blocks on a ten channel system. The output of the first five adder/latch units can reach  $5 \times 64 = 320$  (assuming a 6-bit output on each detector at each  $T_1$ ). The carry out from the fourth ALU denotes a sum above 255 or  $2^8$ . This is fed to the ninth bit of the fifth ALU. The twelve bits on the last ALUs are sufficient to handle the shift and add sums without overflow.

This detector design emulates a CCD type device, but can operate at much higher speeds. The present design, when using high-speed ECL chips should be capable of reliable 40 MHz operation (i.e. an add and loading the latches in 25 ns). This type of design also allows us to reset the data in the adders, to simply shift the data out, or to parallel dump the output to a second shift/add array. With a regular CCD, the input (light) must be 0 when data is to be shifted, otherwise an add of new data also occurs. In our case, we choose when to sample the detector A/D and can thus set up data in the cells while the shift is being performed to ease timing restrictions. These functions are included in our detector and are difficult to perform with a CCD device. They are useful for a lab test system which uses various types of data encoding methods. This detector design is also attractive because it permits several versions of the basic optical architecture and several number representations to be studied without the need to re-fabricate all of the external hardware. It also has the advantages of speed and that we can change to different types of detectors easily.

The output of the  $P_3$  system can be fed into a mixed-binary to binary converter. This unit (Fig. 6-3) can easily be fabricated, but is presently performed in software. The device can handle radices greater than two although we describe its use for binary data representation. The output of the shift/add array is a word (each  $T_1$ ) that can be the sum of 10 6-bit numbers, which has a maximum value of 640. Thus, the output words will be a maximum of ten bits wide. These  $2N-1$  words (produced sequentially in  $T_2 = (2N-1)T_1$ ) represent the  $2N-1$  values of the convolution of the two  $N$  digit data streams. These are encoded as  $2N-1$  sets of 10-bit numbers and correspond to an  $(2N-1)$ -digit mixed radix data representation with each mixed radix digit being binary and a maximum of 640 (i.e. 10 bits). To convert this output into conventional binary representation, refer to the example in Figure 6-4.

The hardware design is quite obvious and would be easy to implement in ECL, and its

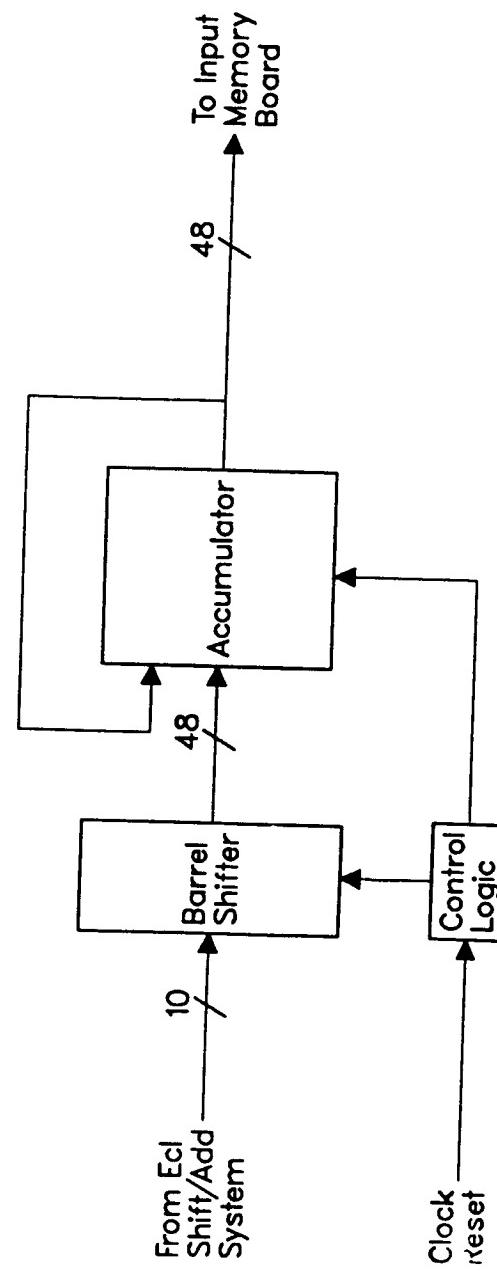


Figure 6-3: Diagram of the Mixed-Binary to Binary Converter

design using 10KH ECL chips has been completed. The advantage to building this device in hardware rather than realizing it in software is that it reduces the output data rate from the shift/add system. This would remove the need for the demultiplexer since the output rate is now less than the rate at which the input memory boards can accept data. The final binary output will have more than 12 bits and thus would require more than one 12 bit input memory channel. This is of no practical concern since the input memory channels are available.

$$\begin{array}{rcl}
 (1357)_2 & = & 7 * 2^0 = 7 = 111 \\
 & & 5 * 2^1 = 10 = 101 \\
 & & 3 * 2^2 = 12 = 11 \\
 & & 1 * 2^3 = 8 = 1 \\
 \\ 
 & & \hline \hline \\
 37_{10} & 100101_2 & = 100101 \text{ Binary}
 \end{array}$$

<b>Mixed-Binary output from Shift/Add</b>	<b>Operation of mixed-binary to binary Converter</b>	<b>Output to memory boards</b>
---	--	------------------------------------

**Figure 6-4:** Operation Performed by Mixed-Binary to Binary Converter

### 6.3 Host Computer System

The Matrix-Vector hardware consists of various blocks that are discussed in this section. This modular approach allows sections of the system to be altered or improved easily. It also allows the user to decide which blocks are needed for a given optical architecture or application.

The system consists of three main parts: the host computer system, the electronic support hardware and the optical system as shown in Figure 6-5. The host computer system consists of the computer chassis, the disk chassis and the tape drive. The optical system consists of laser diodes, AO cells, fiber optics, lenses and other such components. The remainder of the system is

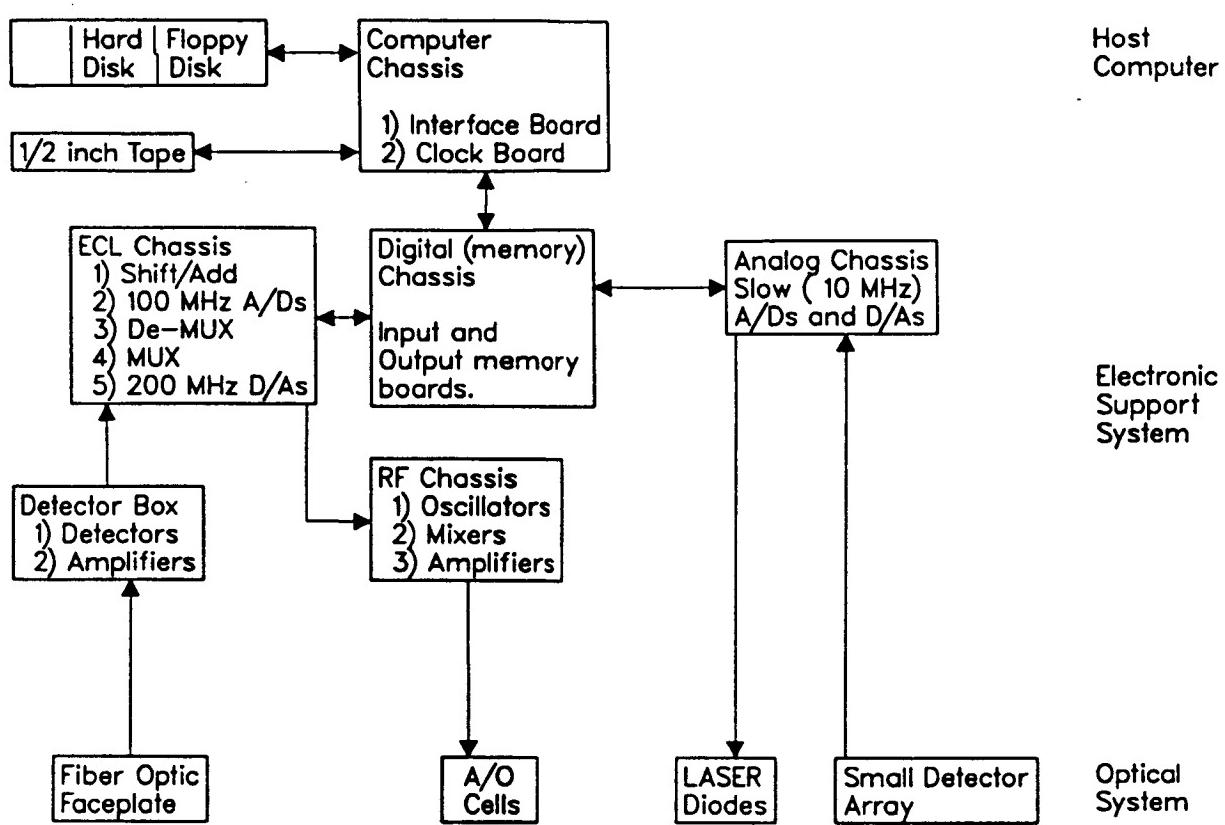
the electronic support hardware. The memory subsystem includes an interface board and the clock board in the computer chassis and the memory boards all of which are in a separate multibus rack. The other support hardware includes various detectors, amplifiers, data conversion circuits and driver circuits.

The host computer supplies data to and collects data from the optical processor via the support hardware. When a Matrix-Vector operation is encountered, a subroutine is called that loads the output memory boards with the matrix and vector data. The memory sub-system is then started and it sends the data to and collects data from the optical system. The memory system indicates to the host that it has completed running and that the processed data is ready. The use of these buffer memories allows the system to be tested at full speed although the processing actually occurs in bursts. The computer system and memory boards are now discussed more fully.

### **6.3.1 Computer System**

The computer used to control the optical system is an Intel 286/380 computer running the iRMX 86 (Intel Real-time Multitasking eXecutive, 8086 processor) operating system. This operating system and computer were chosen since they are well supported, compared to the Pacific Micro 68000 (PM68K) used with our prior frequency-multiplexed optical system<sup>5</sup>. It also is capable of executing software from either our PM68K UNIX system or our 11/750 VAX/VMS with little or no modification.

The computer is housed in an Intel chassis that includes a power supply and a twelve slot Multibus I card cage. A second chassis includes a 32 Mbyte hard disk and a 1.2 Mbyte 8 inch floppy disk. There is also a 1/2" tape drive unit in the rack. We presently have two terminals and a Printronix line printer attached. A third serial line is hooked up to our VAX 11/750 for file transfer. A photograph of the system is shown in Figure 6-6 and a board list is shown in Figure 6-7.



**Figure 6-5: System Block Diagram**

The system runs Intel's iRMX operating system. The present configuration supports two terminals, one printer, a VAX link and one megabyte of RAM. The system presently has Intel's Aedit editor for program generation. The languages on the system include the ASM-86 assembler, PL/M-86, iC-86, Fortran-86 and other system utilities. The system has an extensive amount of diagnostic programs to help with maintenance.

Most of the matrix-vector software is written in C to utilize the software previously

ORIGINAL PAGE IS  
OF POOR QUALITY.



- ← Power Supplies
- ← ECL D/A and Shift/Add
- ← Memory Boards and 10 MHz Analog Rack
- ← Card Cage Chassis
- ← Disk Chassis
- ← 1/2" Tape drive

**Figure 6-6:** Photograph of the Intel 86/380 System

developed for the PM68K system that ran the frequency-multiplexed optical system<sup>5</sup>. This also would make a change to UNIX easier if it were to happen. Because of the 80287-8 floating point processor, most C programs run faster on the Intel system than on the PM68K.

Slot Number	Board Description	iSBX Card	Priority
J01	iSBC 215 Disk Controller	iSBX 218 Floppy Controller	1
J03	CPC Tapemaster A		2
J05	iSBC 86/30 Processor – Not installed		3
J07			
J09			
J11	Multi-interface Board		
J13			
J15	iSBC 286/10 Processor Board	iSBX 354 Serial Card	8
J17	iSBC 012CX 512K RAM Board		9
J19	iSBC 012CX 512K RAM Board		
J21			
J23			
J25	Clock Board		13
J27			14

**Figure 6-7:** Board Layout for Intel Multibus 1 Chassis

### 6.3.2 High Speed Memories

The high speed memories are used to send data to and take data from the optical system at the rates necessary to test the speed of the system. These boards were designed and built at the Center for Excellence in Optical Data Processing specifically for this purpose. They are quite general though and could be used for any application that requires large amounts of input or output data at high speeds.

The interface card can be placed in any slot in the Multibus 1 chassis. The P2 connector on the interface card is extended to the rack containing the memory cards as shown in Figure 6-5. This card contains address decoding logic for the memory cards. It also controls the speed at

which the system runs by dividing down either an internal 20 MHz oscillator, or an external oscillator, by two and then using a programmable counter to divide this clock by one to sixteen. The divided clock is then sent to three circuits that individually delay the clock for data to be sent to or from the  $P_1$ ,  $P_2$  and  $P_3$  sections by small intervals and supplies the memory boards with these delayed clocks. This allows the user to produce slight time adjustments between the data in both AO Cells and the detector plane. This is necessary since there is a delay associated with the AO cells (5-800 ns) and since the physical distance between modulators and detectors is large (5-10 ns for optical propagation of light).

This card is accessed through five 8 bit I/O ports. One port is used to set the clock speed and to set and check the board status. The other three ports are used to delay the clocks with the lower four bits being a fine adjustment of about 4 nanoseconds and the upper four bits being a coarse adjustment approximately equal to the period of the master clock, usually about 50 ns (20 MHz clock).

We presently have six memory cards configured as four output cards and two input cards on the Intel system. There are also two output cards and one input card on the PM68K. Each memory card contains eight data memory channels 4096 words long and 12 bits wide. There are also two sequence control channels on each memory card that allow simple looping operations and are also 4096 words long. The memory boards run at a 10 MHz rate per channel. The memory cards are accessed in I/O space on the main system by setting an address counter port and reading or writing a transfer port. Another I/O port allows the boards to be reset and allows one to change certain status fields particular to each memory board. When a board encounters a stop instruction in the sequence memory, it signals the interface board via a DONE line.

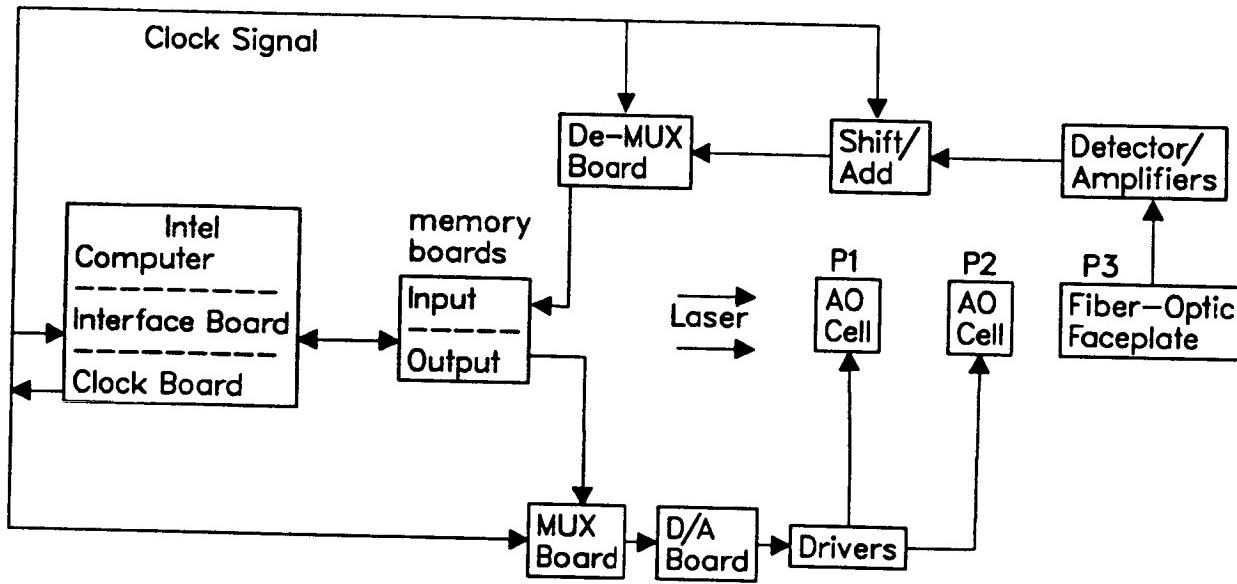
## 6.4 Multi-Channel Encoded Processor Hardware

The analog hardware for this optical Matrix-Vector system is now discussed. This hardware was built to provide the high data rates the system requires. The high-speed analog hardware is shown in Figure 6-8. It consists of:

1. A clock board to provide all the components with the proper clock frequency at the proper time.
2. Multiplexers (MUX) to provide the 40 MHz data rate needed from the 10 MHz memory boards. (These boards were recently constructed and are presently being tested)
3. Very fast D/A converters (4 bits @ 200 MHz, 3 bits used) to drive the AO cells. (One per AO channel)
4. An RF driver card for each AO channel.
5. An oscillator shared by all drivers to provide input data to both AO cells at the correct center frequency.
6. Optional secondary oscillators to allow for frequency-multiplexing of additional data at other center frequencies.
7. A faceplate consisting of 10 (presently 3 exist) precisely aligned SELFOC lenses feeding fiber optic cables to couple the light to the detectors.
8. A discrete detector array consisting of 10 amplified photodetectors and amplifiers to control and adjust the gain and offset of the detector output. (One per detector)
9. Inverting buffers used to make the positive going output from the detector box negative going to be compatible with the A/Ds used. (One per detector)
10. 100 MHz 6 bit A/D boards to digitize the photodetector outputs. (One per detector)
11. A precision reference supply for the A/D converters. (not shown)
12. A Shift and Add board used to emulate a high speed CCD type output detector device in hardware.
13. A demultiplexer to reduce the 40 MHz data rate from the shift/add board to the 10 MHz rate of the memory boards. (Not included at this time)

Many of these parts have capabilities that exceed what is needed for their respective

functions. This occurred since some of the parts were available or because the faster parts were easier to work with. Each of these elements is now discussed.



**Figure 6-8:** High Speed Analog Hardware, Block Diagram

#### 6.4.1 Clock Board

The clock board was built on a multi-interface board similar to the interface board for the memory boards. This is a Multibus card with all necessary control logic and a prototyping area on it. The board was built to allow the system's operating frequency to be changed easily under program control. The board has three ECL oscillators with frequencies of 100, 80 and 10 MHz.

The outputs of the oscillators are fed into a 4-1 multiplexer controlled by an I/O port on the computer. This multiplexer selects which oscillator is to be used. The output of the multiplexer is then fed into a ECL counter that provides  $\div 2$ ,  $\div 4$ ,  $\div 8$ ,  $\div 16$  outputs. The multiplexer output is also fed into two other 4-1 multiplexers along with the  $\div 2$ ,  $\div 4$ ,  $\div 8$  outputs of the counter. One of these multiplexers is used to feed an ECL-TTL converter and send a clock signal to the high-speed memory interface board. The other multiplexer sends a signal to two programmable ECL delay lines, which are used to adjust the timing of the multiplexers, shift/add and demultiplexer boards to be compatible with the memory boards and with the optics. The outputs of the delay lines are buffered and sent to the ECL boards differentially to minimize noise problems. Since we are presently not using the multiplexers, the shift/add board obtains its clock signal from the input memory board via the cable it uses to send data to the memory board.

#### **6.4.2 Mux/DeMux Board**

Since the system presently built does not require data rates higher than 10 MHz and since fast ALU chips were not available when the system was constructed, the Mux/Demux system has not yet been fabricated. If we build larger systems that use multi-channel AO cells in  $P_1$ , data must be fed to the optical system faster (and with more channels of data) in order to utilize the information presently in the  $P_2$  AO cell.

The design recently constructed consists of TTL-ECL converters, ECL 4 to 1 multiplexers and ECL differential drivers (OR-NOR gates). The board will derive its timing signals from the clock board. To increase the speed we consider each 12 bit memory channels as four 3-bit words of data and we use the multiplexers to switch between these words. This gives us a three bit output at four times the input rate. The only anticipated problem is obtaining the correct timing which should not be too difficult using the clock card.

#### 6.4.3 ECL Shift/Add Board

In order to perform the convolution needed in the DMAC algorithm used, we use a time-integrating architecture. Since a CCD type detector of the required capabilities is not available at this time, we implemented the same function in digital hardware. This board performs that function by taking the output from the A/Ds and adding that data to the data stored from the previous add. This method also gives us certain advantages such as being able to monitor the data at any point in the output system.

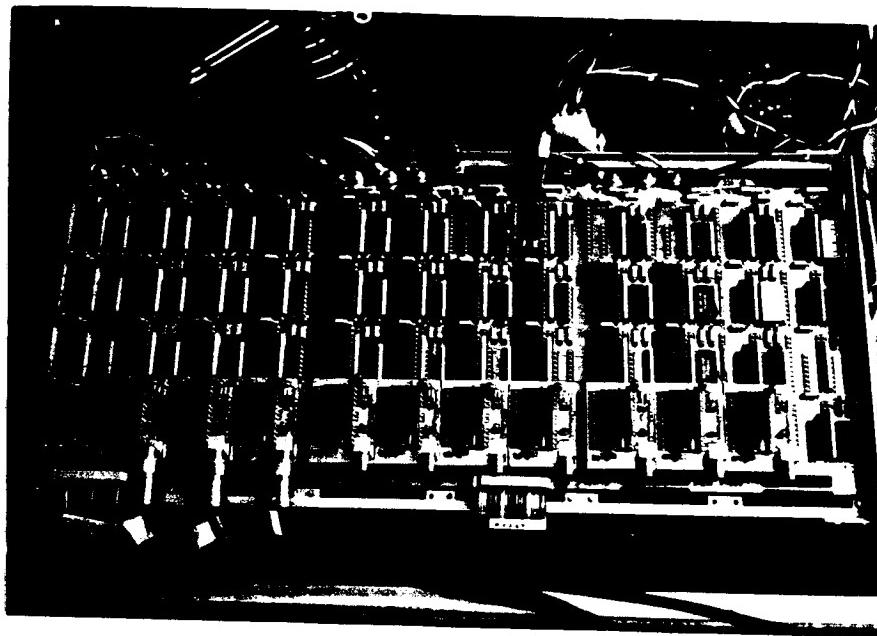
This board was built on a special ECL wire-wrap panel<sup>26</sup> designed for speeds up to 100 MHz. A two channel board was wire-wrapped locally to test the design and the ten channel version was wire-wrapped by Augat using a special program that made sure twisted pairs were used for wires longer than a critical length and that all lines were properly terminated. The program also provides a large amount of information about wire lengths and other parameters that are helpful in debugging the board. A photograph of the board is shown in Figure 6-9.

This board has nine adder-latch blocks and some discrete logic to control their operation. It consists of ALU's and latches designed to emulate a CCD array, but at much higher speeds. The latches are used to perform a shift of one location per clock cycle. The shift/add array can perform three basic operations. These are:

1. To feed data from the A/Ds directly to the latches. (RESET)
2. To add the new input A/D data and the data in the prior latch.
3. To shift data from one latch to another through the ALUs.

The control section logic for the shift/add system is shown in Figure 6-10. It operates from two 4-bit counters wired as a 5-bit counter. This gives a counter with a maximum output of 32 and allows for a maximum of 32 cycles or operations (i.e.  $32 T_1$ ). The number N of cycles ( $T_2 = NT_1$ ) is set by a 5-bit comparator fed with the counter data lines as one set of inputs and 5 position

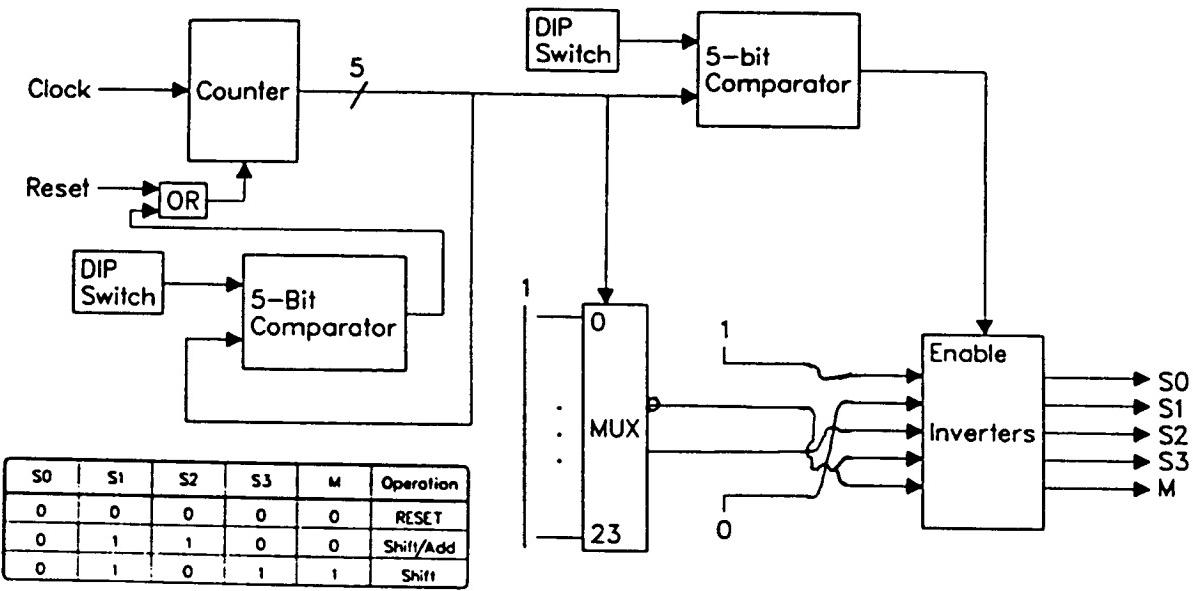
ORIGINAL PAGE IS  
OF POOR QUALITY.



**Figure 6-9:** Picture of the ECL shift/add board

DIP switch data as the other inputs. When the comparator indicates that the outputs are equal, a reset signal is sent to the counter to reset it to zero. This reset signal is different from the RESET signal used as a control signal to the ALUs.

The ALUs have five inputs that control the function they are to perform. Since only three functions are needed as enumerated above, the control logic is somewhat simplified. The control lines on the ALUs are fed to inverters with enable inputs. When the enable line is low, all the inverter outputs are forced low. This line is controlled by a 5-bit comparator, allowing the RESET operation to occur on only one of the 32 possible cycles. The inputs to the inverters are either forced low or high, or are controlled by a set of multiplexers. The multiplexer inputs are forced high or low depending on whether the current cycle is to be a shift or an addshift



**Figure 6-10:** Control Section Block Diagram

operation. Since the A/Ds do not produce valid data until 7 ns after the convert clock has occurred, the control section of logic has 7 ns to generate the proper signals to tell the ALUs what function to perform.

A multiply can use from one to 32 of the basic cycles. For the case  $N=10$  (10 digit multiply), there are 19 cycles required since all  $2N-1=19$  outputs are needed. A cycle here is one  $T_1$  time and the full multiply requires  $2N-1$  cycles. In cycle 0, the counters on the board are set equal to 0. Our standard setup uses the first two cycles (0 and 1) to finish shifting out the data

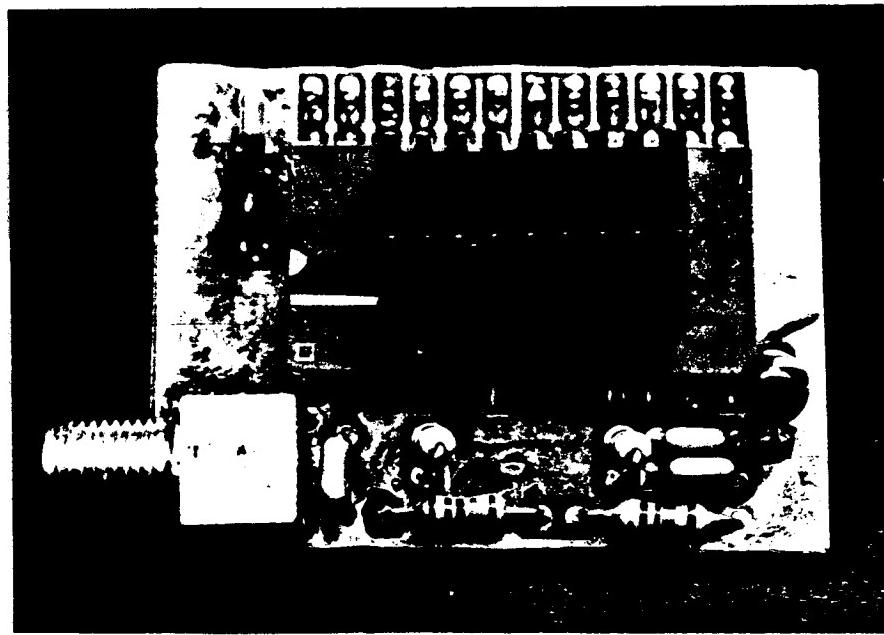
left in the latches from the previous multiply to the input memory board. This is required since a sample at time 0 on the A/D appears on the A/D output pins two cycles later. On cycle 2, we perform a RESET operation which loads the outputs of the A/Ds into the latches through the ALUs. An add/shift operation is done for the next N-1 cycles. With 10 detectors this is 9 add/shift cycles. We then perform a shift for N-3 cycles to output the data remaining in the latches. We use N-3 cycles since two shifts are performed in cycles 0 and 1. These N-1 shift cycles could be avoided (or pipelined) by using a second set of latches. The second set of latches are not included on the present board. For a two's complement system, this second set of adder/latches would be useful. However, we plan to use negative base number representation techniques that do not require this second set of latches. The ECL shift/add board outputs a 12 bit (with up to 10 bits significant) ECL signal at the fast clock ( $T_1$ ) frequency.

The clock circuits that control the adders have been tested at speeds up to 70 MHz. Decoding delays in some of the logic, plus the time for an add (worst case approximately 15 ns) place an upper limit of about 40 MHz on the complete circuit. We tested the circuit design at 10 MHz using presently available 10K ECL adders (add time approximately 24 ns). We can upgrade the device using fast ALUs that are now available. As noted earlier, the multiplexer is not used or needed at present, although it is being worked on. The 10KH parts operate with half the propagation delay of the regular 10K ECL parts with no increase in power requirements.

#### **6.4.4 100 MHz 6-bit A/D Boards**

These A/D chips exceed what is presently required by the rest of the system except for the D/As. These parts were used because they were available. The excess capacity is useful since we have more confidence that the A/Ds will work correctly at the data rate that they are being fed. Alternatively, we could have purchased 40 MHz A/Ds that have a higher resolution (about 8 bits). This would allow us to use more channels or higher radix numbers.

Each A/D board has one TRW 1029J 100 MHz 6 bit flash A/D converter with an impedance matching network as shown in Figure 6-11. The impedance matching network is used to terminate the input with a 50 ohm load and to provide the A/D with a 18 ohm source as specified on the TRW data sheet<sup>27</sup>. The board is designed to be inserted into a 24 pin, 0.6" wide socket on the shift/add card which also supplies the necessary clock pulses, references and power supply voltages. Each A/D board is designed with a large ground plane to help minimize noise. All supply and reference voltages are decoupled on the board as close to the chip as possible.



**Figure 6-11:** Photograph of one 6 Bit, 100 MHz A/D board

#### **6.4.5 A/D Reference Supply**

The reference voltages for the flash A/D converters are furnished from an external supply board. The flash A/D converters require one volt across the resistor ladder with the R<sub>b</sub> (Resistor ladder bottom) voltage of -0.3 V and an R<sub>t</sub> (Resistor ladder top) voltage of -1.3V<sup>27</sup>. The board uses an LM368-10 precision 10 volt reference. This supplies inputs to two LH0021CK 1-amp op

amps which have the proper resistor values to supply the above voltages. All supplies are bypassed by  $0.1 \mu\text{F}$  capacitors at each chip and by  $12 \mu\text{F}$  tantalum capacitors on the supply lines. The feedback resistors on both supplies include a 500 ohm 15-turn trimpot to allow the outputs to be precisely adjusted within specs to set the range and gain if necessary. The outputs are sent via a shielded cable to a DIP header where they are sensed remotely. In testing, the board exhibited excellent noise and regulation characteristics with the noise on the -0.3V side about 1 mV and the noise on the -1.3V side unmeasurable. The noise level with the system running is on the order of only a few millivolts and is not expected to cause any problems.

#### **6.4.6 Four-Bit 200 MHz D/A Converter Boards**

The four-bit D/A converter board (built specifically for this processor) consists of eight D/A chips each capable of running at 200 MHz with three converters per chip<sup>28</sup>. This gives us 24 D/As per board. These will be used as two groups of 12 D/As (4 chips) with 10 of the 12 D/As in each group used. This is done since we will be using up to 10 channels on each of the two AO cells, i.e. with one D/A for each of the  $N+M=20$  possible AO inputs. Ten channels were decided on to keep the system size reasonable while still allowing us to test most of our architectures. The board also has ECL receivers to condition the input data before it is fed into the converters. This makes the D/A board very flexible since it can be driven by any system capable of driving TTL-ECL converters or ECL OR/NOR gates. Also, by using the differential drivers, the system has a higher noise immunity and can be many feet from the driving circuit even at very high speeds. The D/A converters are designed for video use and have various signal inputs such as sync that are not used in this application, but the board was designed to allow them to be utilized. An example could be the need to move the output voltage levels down by 0.03 volts (-0.03 to -0.63 into 75 ohms) which can be done with the BRIGHT control input to the board. Each converter also has a clock input and latches to buffer the data so as to reduce glitches.

If the multiplexer is put in use, we will only be using the top three bits of the 4-bit D/A converter. This multiplexer is designed to use only three bits. This was done since we desire a 40 MHz rate in the multi-channel system, whereas if we used four bits the data rate would only be 30 MHz from the 12 bit, 10 MHz memories. We presently use all four bits on the initial test system since there is no multiplexer. These D/As could also be used with different multiplexer designs to provide 4 bits of data at 200 MHz if needed. Using four bits gives an output voltage swing from 0.000 to -0.450 volts into 50 ohms. Since the D/A converters have internal 75 ohm resistors, it was necessary to use external 150 ohm resistors to match the line impedance (50 ohms). Future versions of the D/A converter are planned that will allow for higher voltage and current levels when driving a 50 ohm line. When these parts become available and if a greater voltage swing is needed to drive the RF mixers, these units should be directly compatible with the present PC board layout. Sample outputs from the 4-bit D/As are shown in Fig. 6-12.

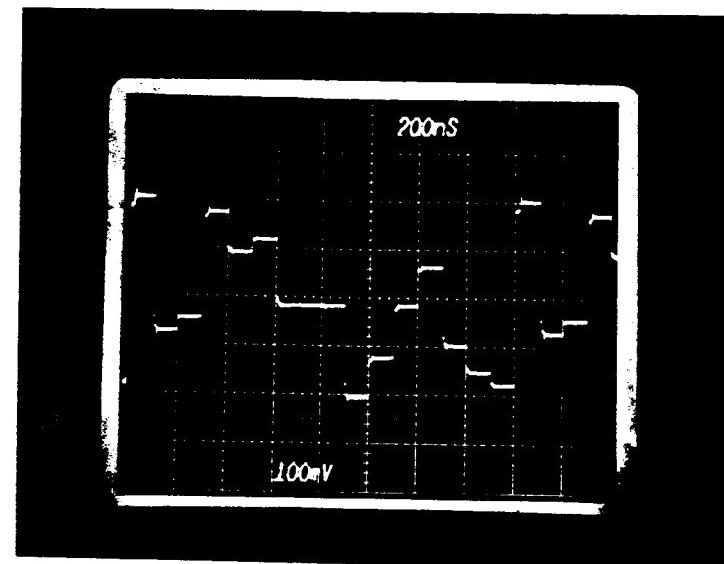
#### **6.4.7 RF Drivers and Oscillators**

The RF driver boxes contain local oscillators, splitters, mixers, combiners and amplifiers. The actual design of the box can be separated into two sections, the oscillator board and the driver boards. The oscillator section provides the RF frequencies that are used to drive the RF inputs on the mixers. The driver boards have the circuitry necessary to modulate the RF input, perform frequency multiplexing and amplify the resultant signal to be able to drive an AO cell.

We have oscillators with frequencies of 300.000000 and 400.000000 MHz. We use two frequencies since one of our 32 channel cell operates at 300 MHz and the other at 400 MHz. Each oscillator is amplified and fed into a splitter network to provide multiple outputs with the same frequency and phase. This part of the system uses SMA style cases and coaxial cable to interconnect all the components.

The outputs of the oscillator section and the D/A section are used as the inputs to a Level

ORIGINAL PAGE IS  
OF POOR QUALITY.



**Figure 6-12:** Four bit D/A generating a random output

13 RF mixer. The Level 13 refers to the LO input specifying that it should be +13 dbm. Each driver board has room for three mixers with two mixers presently installed. This was done to allow us to handle complex data that needs two-channel frequency-multiplexing. The outputs from the three mixers are fed to a three-input combiner. The output of the combiner is fed to an impedance matching network since an CATV amplifier is used that needs a 75 ohm input and the mixers are 50 ohm devices. The matching network is needed to extract the full performance of the amplifiers and to reduce excessive power dissipation (heat). The amplifier output is then fed through another matching network to match the 75 ohm output with the 50 ohm AO cell's input impedance. The driver cards are built on a PC board using stripline techniques.

## **6.5 Detector Array and Fiber Optic Coupling**

The optical system's outputs are piped by fibers which are then connected to the detector system. The detector array consists of ten Merit 1900 hybrid photodetectors capable of 100 MHz operation, each with an integral pre-amplifier on the hybrid as shown in Figure 6-13. The pre-amplifier (detector) output is then amplified by a high-power, wide-bandwidth amplifier capable of driving a 50 ohm line. The detectors are all mounted on a single copper heat sink in order to keep them all at approximately the same temperature. This is needed so that output variations due to detector temperature are somewhat constant over all the detectors. The detectors are very sensitive and exhibit a drift of about 10-20 mV/ ° C. Since the output is amplified with DC coupled amplifiers, this drift is made worse. We counter this by leaving the detectors on at all times and attempt to keep the room at a fairly constant temperature.

The amplifiers on the card are of a hybrid type made by Comlinear. They have adjustments for gain, offset and crossover. The gain and offset are fairly standard adjustments except that the gain is somewhat affected by the crossover adjustment. Since the Comlinear amp has a temperature drift that effects the gain at low frequencies, a second op-amp is included on

the board to handle lower frequencies and it stabilizes the gain in this region. The crossover adjustment is used to control the region where the second op-amp has an effect. The recommended gain setting procedure is to feed the input of the circuit with a 70 kHz square wave and adjust the crossover adjustment for best symmetry. Since we have a gain trimpot installed that can vary the gain of the amplifier over a wide range, some resistors have to be changed once an approximate gain setting is made to put the crossover control in the correct operating region. The proper initial adjustment procedure for the detector amplifiers is to first feed in a normal operating signal pattern and adjust the gain control for the desired output. A good program to use for this is in the :prog:c/setone directory called CPP. It outputs a pattern with a single pulse, three off-cycles, three on-cycles and then 6 off-cycles. This pattern repeats every 13 cycles. We then run a program in the same directory called SEVENTY that feeds the system with a 70 kHz square wave to adjust the crossover. If the trimpot does not have sufficient travel, the operator must calculate a new resistor value based on the Comlinear data sheets. The proper offset is about 0.43 volts to turn on the least significant bit of the A/D. Since we are presently using the third bit, the easiest method to adjust the offset is to monitor the A/D outputs on the logic analyzer, then run a program such as the CPP routine mentioned above and adjust the offset until only the least significant bit used (bit three) is switching.

The detectors used are physically large, compared to a CCD or an integrated detector array. In order to couple the output light to the detectors, we use a fiber-optic setup to pipe the light to the detectors. This is achieved by a faceplate with SELFOC lenses and fibers placed in the P<sub>3</sub> output plane shown in Figure 6-14. The SELFOC lenses are the small rod elements at the front of the metal holding block. These are attached to the block first. Then the fibers are butted up against the SELFOCs, aligned with a laser and attached. The stack of metal plates on the back of the assembly are used as strain reliefs. A SELFOC lens is a thin rod of a lens type material that has been doped such that its refractive index varies radially. This makes the rod

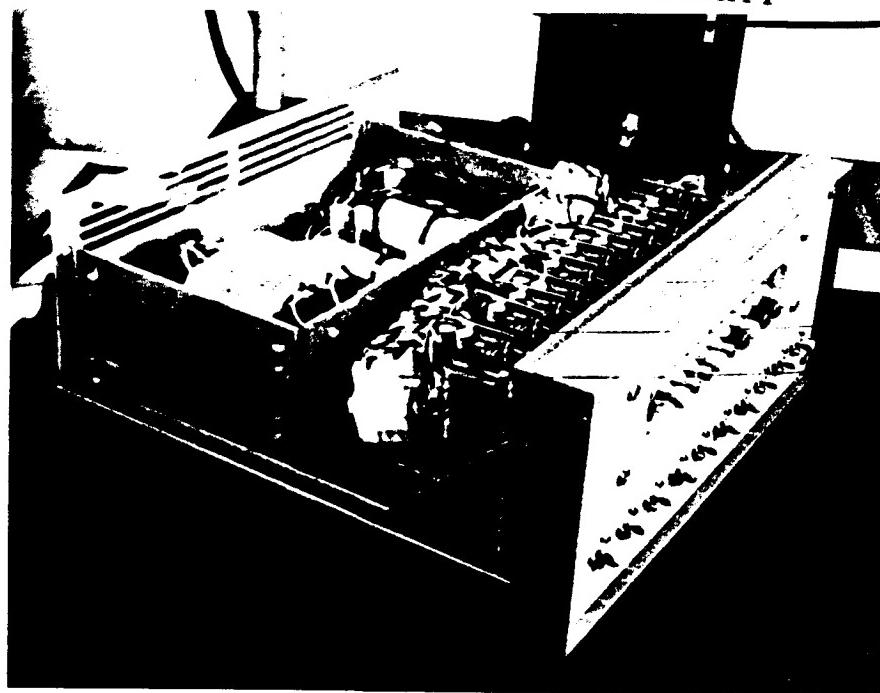
act like a lens even though it has flat surfaces. These SELFOC lenses are coupled to fiber optic cables that have a connector on the other end designed to plug directly into the detector case. The SELFOC lenses used are 1 mm in diameter, 2.55 mm long and have a pitch of 0.25. The coefficient of refractive index distribution is  $0.6158 \text{ mm}^{-1}$  at 630 nm. The fibers used to connect the SELFOCs to the detectors are a multimode type with a core size of  $100 \mu\text{m}$ . The SELFOCs are placed on 2 mm centers. The present version has some alignment problems (discussed under system construction in sections 6.9 and 6.10). We have recently found new methods of fiber coupling<sup>29, 30</sup> and plan to use these in the ten channel system.

## 6.6 Software

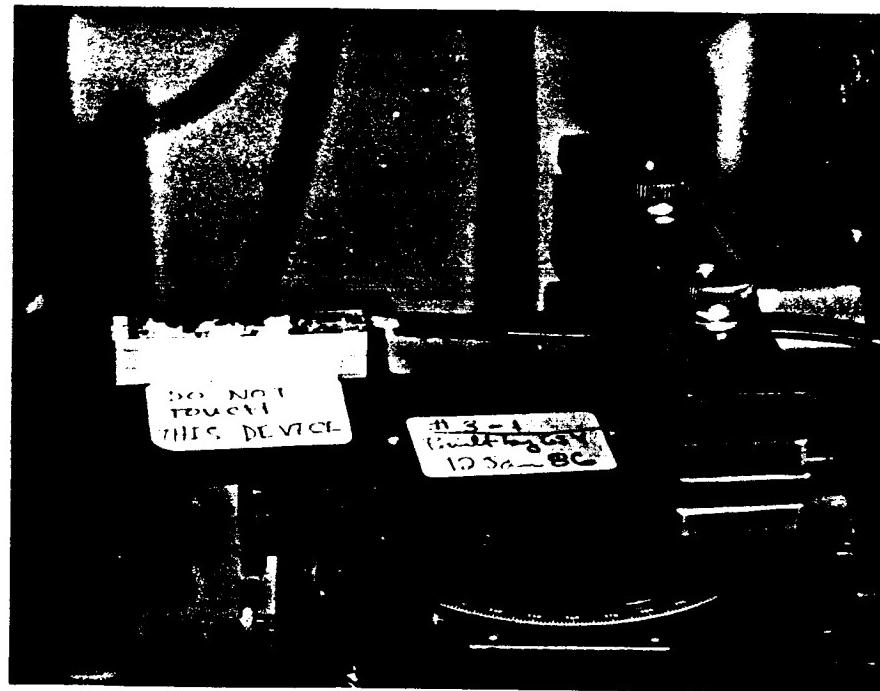
This section describes the software written to operate the Matrix-Vector processor. This software consists of an assortment of low level routines that handle loading of the high speed memory boards, converting bases, decoding the detector outputs and various other functions. A multiply routine is then written with the low-level software. The low level software is detailed in Section 6.7. The top level of software consists of modified versions of LAE solutions written to solve finite element or similar problems.

All software is presently written for a single input  $M=1$  channel system with  $N=3$  channels at  $P_2$  and an LU solution of the LAE. We perform all multiplications with 21 bit input accuracy. The software has most of the necessary controls to be able to operate larger optical systems. It also has the capability to handle bipolar numbers and varying bit widths. Most of the basic software is common to any LAE solution, iterative or direct.

ORIGINAL PAGE IS  
OF POOR QUALITY



**Figure 6-13:** Photograph of the detector box used



**Figure 6-14:** Photograph of the SELFOC/fiber-optic setup

## 6.7 Low Level Routines

These subroutines are designed to make it unnecessary for the user to be concerned with hardware details. They take care of all the memory board functions very efficiently with the code written in C for compatibility. They also determine where data is to be placed on a memory board and they perform base conversions.

### 6.7.1 Data Handling Conventions

The software must be able to handle bit lengths longer than the number of  $P_2$  channels used and it also must be able to handle fractional numbers. The longer bit lengths require more time, but show the flexibility of the system to be configured to run any number of bits dynamically. We now show how this is done on our  $N=3$  channel demonstration system.

Step 0 in Figure 6-15 shows the operations on a full system to perform a 9-bit multiply on a system with  $N=3$  channels at  $P_2$ . We feed the three least significant bits (LSBs) of the multiplicand to  $P_2$  and run the 9 bits of the multiplier into  $P_1$  as shown in step 1 of Figure 6-15. We then feed the next three bits of the multiplicand to  $P_2$  and repeat the 9 bits of the multiplier at  $P_1$ . This sequence is continued until the 9 bits of the multiplicand have been fed into  $P_2$ . The final output is assembled by shifting the second output product (from step 2) by three bits and adding it to first product (from step 1). The third and successive output products are shifted by 6,9,12 etc. bits and added to the previous total. The proper ordering, shifting and handling of the data and these steps are performed in the multiply, memory load and unload routines. This method thus allows us to perform a multiply of any length on an optical system of any length. It is obviously preferable (from time considerations) to have  $N$  large, but with this method, we can easily trade accuracy and hardware costs for speed (depending on the problem being performed).

The actual time for a multiply to be performed on the lab system is therefore,

Step 0		
	101101101      (365) 001011100      ( 92) -----	---
Step 3	Step 2	Step 1
101101101 001 -----	101101101 011 -----	101101101 100 -----
101101101 -----	101101101 101101101 -----	10110110100 -----
101101101 >----	1112112111 >--    -> 1112112111  -----> 101101101 -----	10110110100 1112112111 101101101 -----
		102223323221100 (33580)

Figure 6-15: 9 Bit Multiply on an 3 Bit Optical System

$$(B/N) \times (N + B - 1) \times T_1 \quad (6.1)$$

where the number of channels in  $P_2$ ,  $N$ , on the lab system is 3 and the bit length,  $B$ , is 21. The previous chapters that define  $T_2$  as being  $2N-1$  cycles (each  $T_1$  long) consider the system to be performing an  $N$  bit multiply on an  $N$  bit system. The  $T_2$  for the product of each  $N$  bit word by a  $B$  bit word is  $N+B-1$  cycles (each  $T_1$  long) in the lab system since that is the length of the convolution being done.

In an LAE solution, the equation to be solved is,

$$\underline{K} \underline{x} = \underline{p} \quad (6.2)$$

In the finite element problem and many others we intend to solve, there are many purely

fractional numbers. Since the optical system handles only integers, a method must be found to express fractional numbers. Input fractions can be handled by scaling K and p up by a scale factor. This will not change the x result, but will allow us to handle fractional inputs. The next issue is that the solution vector x may also contain fractional values. The solution vector does not change when the inputs K and p are scaled. Any purely fractional output x, would therefore be truncated to 0 and this can cause significant errors. One solution considered was to scale p by an additional factor. This would just add a scale factor to the output that could be adjusted later. Unfortunately, this will increase the dynamic range needed to represent p. The solution chosen is now discussed.

In standard binary notation the input numbers for our finite element problem cover the range of numbers shown in Figure 6-16 with an assumed decimal point after the 7th bit and with 21 bits of accuracy. The range of input numbers for the initial problem spans about  $2^3$  to  $2^{-13}$ , or a range of  $2^{17}$ . Thus a 17 bit processor should be adequate for input representation. We will allow 21 bit computations to accommodate the larger range of values that result from multiplication. In addition, in LU decomposition, the values in one column of the decomposition matrix are divided by the diagonal element (the largest element in a column). Thus, LU decomposition will generate more smaller valued numbers. Hence, we will use the assumed decimal place after the seventh digit and allow seven integer bits and fourteen fractional bits as shown in Figure 6-16

The input vectors are represented by floating point variables in the host system, these numbers are scaled by  $2^{(\text{number of decimal places})} = 2^{14}$  for our case in Figure 6-16. This yields a number representation as an integer variable with a properly scaled binary 21-bit representation. A multiply is performed on the optical system as if the two input numbers are just 21-bit integers. All of the convolution output bits (41) are retained. The assumed decimal point in the

output is located at twice the number of bits that were to the right of the decimal point in the original input data, i.e. at  $2 \times 14 = 28$  bits to the left of the least significant fractional bit as shown in Figure 6-16. The integer part consists of the other 13 bits. We then truncate the output to be the same number of bits (21) as our input with 7 bits to the left of the decimal point and 14 to the right. In doing this, we first discard the 14 least significant bits. We then check to see if the integer portion is greater than  $2^7$  which would indicate an overflow. If an overflow occurred, we set the value to the largest possible number in our notation. In the optical system, we compute all  $2N-1$  output bits and then discard the low-order bits and perform this overflow check. Since scaling is very problem dependent, it is performed in the high-level software such as the LU decomposition routine.

In standard binary notation the input numbers for our FE problem range:

from 8.3770000 = 0001000.01100000100000  
to 0.0002115 = 0000000.00000000000011

Output numbers will therefore have the form

111111111111. dddd dddd dddd dddd dddd dddd dddd dddd  
13 . 28

**Figure 6-16:** Assumed Decimal Point Handling

### 6.7.2 Hardware Dependent Routines

These routines include the functions necessary to load the data and sequence memories, start the processor running and the software to unload the memories. These routines depend on what hardware is present and how many channels of the processor are being used. In order to maintain processor speed, they are fairly specific to the processor set-up. They are also very easy to modify though to use different architectures. The multiply code calls these subroutines in a standard manner.

### 6.7.3 Scalar Multiply Routines

There are four versions of this routine that are obtained by linking with the appropriate object code. Two of the routines perform the multiplications optically and the other two are simulation routines. All four routines are called with the same parameters so the high-level program does not need to be changed to switch between simulation and optical processing. All input numbers are integer long variables (32 bits). One of the variable passed tells the processor what bit length to use for the multiplies. The routines are different as follows,

1. FMULT - This routine uses high level C code to perform the multiplications in floating point. This routine was used to debug the high level code and to verify the results of the other multiplication routines.
2. DMULT - This routine simulates the optical processor in detail. It calculates what numbers would be read from the memory boards and calculates the output from this. This was instrumental in eliminating some errors from C compiler problems. Since the optical system is very reliable, this procedure can be used to simulate the optical processor on any system.
3. OMULT - This performs the multiplications on the optical processor. It calls all of the memory load and unload routines and calculate various control parameters for the system.
4. UMULT - This performs identical to OMULT except that it considers the input numbers to be unsigned. This is used for most of the test and alignment software to verify the optical system.

We are presently modifying the above routines to use multi-level data. Both simulators will now run using 30 bit accuracy with radix 4 (2 bits) encoding. The hardware routines will require some minor re-work in order to allow us to use the multi-level capability. This will increase our accuracy by 9 bits, and speed up the multiply by a factor of two.

Work is also being done on methods to handle complex data either by doing four sets of multiplies or by using the three-tuple method. We are also working on software to use the negative base system to perform signed multiplies.

#### **6.7.4 LU Decomposition Software**

The LU decomposition software performs a direct solution to a LAE. The software also takes into account the bandwidth of the matrix so as to reduce computation time. The user inputs the data on the size of the matrix, the bandwidth and the number of input vectors. The program then reads the input matrix and vector data. It then generates an augmented matrix and performs the LU decomposition and the backsubstitution to obtain the solution vectors. The LU decomposition is performed on the optics since it is the major task and the backsubstitution is performed digitally at this time.

#### **6.7.5 Software list**

Table 6-2 contains a list of all relevant software on the system.

- **MEMORY\_LOAD**  
Loads the data into the electronic interface in the proper form.
- **MEMORY\_UNLOAD**  
Retrieves the output data from the electronic interface.
- **MULT0**  
Passes two real vectors to the optical processor, and returns the VIP.
- **MULT1**  
Digital simulation of an optical MULT0.
- **MULT2**  
Digital simulation of an optical floating-point multiply.
- **CM\_MATH**  
Library of complex matrix operations.
- **FM\_MATH**  
Library of floating-point matrix operations.

**Table 6-2: Table of System Software**

## 6.8 Initial Laboratory System

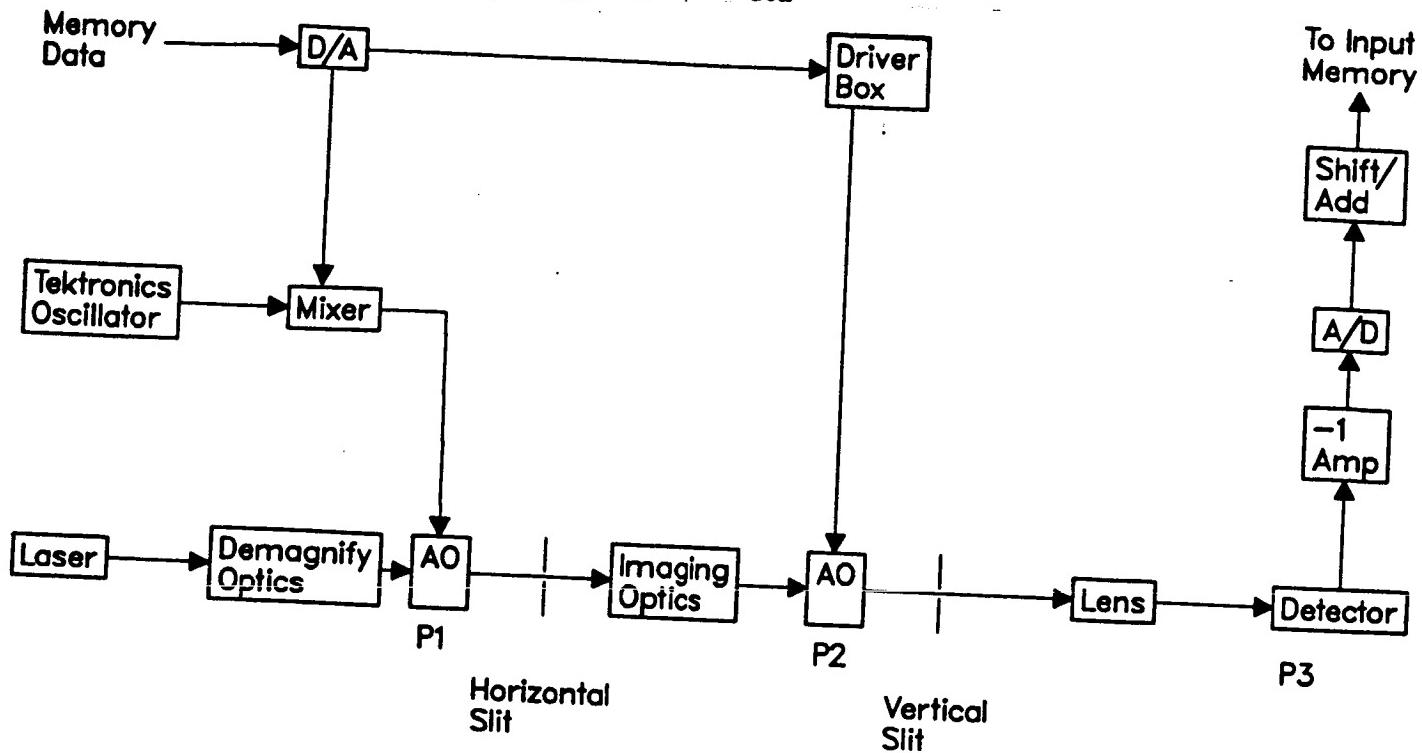
We now describe the construction of the optical laboratory system. The first system, described in Section 6.9, is a single bit system that uses a single channel AO cell in  $P_1$  and one channel of a 10 channel AO cell in  $P_2$ . This system was used to test our ability to align the system timing, to drive the optical system and to verify our light budget. In Section 6.10 we describe the three channel system that is used to test the electronic support system and to demonstrate the use of the optical architecture to solve an LAE. This system uses a single channel AO cell in  $P_1$  and three channels of a 10 channel AO cell in  $P_2$ . This system uses binary data with  $T_1 = 0.1 \mu s$  and a  $T_2 = 2.3 \mu s$  which allows 21 bit multiplies. We will be using the method described in Section 6.7.1 (Figure 6-15), where  $T_2 = (N + B - 1)T_1 = (3 + 21 - 1)T_1 = 23T_1$ . Hence  $N=3$  channels at  $P_2$ ,  $B=21$  bits and  $T_2 = 23T_1$  are used.

## 6.9 Single Bit Test System

### 6.9.1 Construction

This initial system was built to quantify how much light was available and to demonstrate that we could perform a simple product of two binary inputs. When the initial design of the system was completed, there was some concern as to how much light would be available on the detectors after passing through the AO cells and the optics. This system was also used as the test vehicle to align, calibrate and adjust the timing for the optical system. This was achieved by running simple patterns into the AO cells and examining the output on an oscilloscope. The system is diagrammed in Figure 6-17. The system was built using a single channel, longitudinal  $\text{TeO}_2$  AO cell in plane  $P_1$ , one channel of a ten channel, longitudinal  $\text{TeO}_2$  AO cell in plane  $P_2$  and a single detector with amplifier.

The construction of the system proceeded as follows. We first optimized the output of our Spectra-Physics 125 HeNe laser ( $\lambda = 633 \text{ nm}$ ). The maximum output of the laser measured near



**Figure 6-17: Single Channel Test System**

its head was 61 mW. We found that the output power level of the laser would degrade by as much as 50 percent over a week without periodic adjustments. This is due to the extreme 20° F temperature variations in our labs. This is presently being corrected. A more typical variation is 5-10% over a day. This is not significant when running a problem since the laser output is quite stable during the short time it takes to execute a problem. Calibration can be performed to adjust for temperature variations if needed. Our new AC-coupled system should minimize this problem.

The next step was to demagnify the beam to increase the rise time of the output signal from the AO cell in  $P_1$ . The demagnification optics used consisted of a 100 mm lens with a 150  $\mu\text{m}$  pinhole at its focal point and an 18X microscope objective ( $f_L = 10 \text{ mm}$ ) to re-collimate the beam. This reduces the beam width by a factor of 10 from 2.0 mm to 200  $\mu\text{m}$ . A narrow optical beam is necessary since the light beam leaving  $P_1$  is the convolution of the input signal with the

input laser beam. The rise time of the output light is thus dependent on the duration of the  $T_1$  signal and the laser beam width. Since the data travels in the cell at  $4.26 \text{ mm}/\mu\text{s}$ , a  $T_1$  data packet covers a  $200 \mu\text{m}$  distance in approximately 50 ns. Since  $T_1 = 100 \text{ ns}$ , the output light from  $P_1$  is 150 ns in duration (the convolution of  $T_1=100 \text{ ns}$  and the  $200 \mu\text{m}$  laser beam width which corresponds to a 50 ns width in terms of AO cell acoustic velocity) with a rise time of 50 ns, a flat output for 50 ns and a fall time of 50 ns. Demagnification increases the divergence of the output beam. Thus, to keep the size of the beam as small as possible, the AO cell in  $P_1$  is placed as close as possible to the second  $f_L=10\text{mm}$  demagnification lens. This is done because increased divergence will reduce the efficiency of an AO cell, which only gives a large diffraction efficiency for light within a certain range of the Bragg angle.

When the optical beam leaves the AO cell at  $P_1$  (AO1) it diverges both horizontally and vertically. The optical system between  $P_1$  and  $P_2$  consists of three cylindrical lenses which images a magnified version of AO1 onto AO2 horizontally and thus the beam divergence horizontally is not a major concern In the vertical direction the diverging beam from AO1 is focused onto AO2 by a third cylindrical lens. This yields a vertically diverging beam leaving AO2 with a horizontal width at  $P_2$  set by the width of the acoustic channel in AO2. This must now be focussed onto the detector system at  $P_3$ . In the first setup, (this section), this was achieved with a 30 mm spherical lens. In the second setup, (Section 6.10) a fiber optic detector faceplate was placed 10 cm from AO2 and the SELFOC on the detector faceplate served to focus the light into the fiber optics used to couple to the detectors.

The  $P_1$  AO cell is then placed in the beam and is positioned for maximum - 1 order diffraction efficiency. The AO cell used has a center frequency of 200 MHz and a bandwidth of about 60 MHz. This is more than adequate for this system which will only be run at a maximum of 10 MHz. The diffraction efficiency of this AO cell at the power level used is about five

percent. Thus, much of the optical beam power is lost, but sufficient light exists to be quite usable.

The drive circuit for the  $P_1$  cell consisted of a Local Oscillator (LO), a mixer and an RF amplifier. A Tektronix model SG 503 Leveled Sine Wave Generator with a 10 db attenuator on the output and adjusted for a +7 dbm signal was fed to the LO port of a Mini-Circuits ZFM-1W mixer. The IF port on the mixer was fed with the output of the 4-bit D/A circuit described in Section 6.4.6. The mixer output was then fed through a 10 db attenuator to a Mini-Circuits ZHL-1-2W RF amplifier which provided an output of about 50 mw RF to the AO cell.

The next step is to block the DC term leaving the AO cell and expand the -1 order light horizontally so that it illuminates all channels of the second AO cell. The input light distribution is uniform within 5% over three channels of AO2. This is achieved with a 12.7 mm and 200 mm horizontal cylindrical lens magnification system. We then compress the beam vertically to about 400  $\mu\text{m}$  at AO2. This was achieved with a 300 mm cylindrical lens and was necessary since the beam diverges vertically as it leaves  $P_1$ . This is also necessary since the  $P_2$  AO cell has a very narrow vertical opening and the crystal is set back in its case. The compression also provides us with a smaller beam exiting the cell that is easier to image onto the output detector thus increasing the output light detected.

We then place the  $P_2$  AO cell in the system and adjust it for the correct Bragg angle by checking for maximum output. This cell has a bandwidth of only 10 MHz, so it is just able to handle the input signals we intend to use. In this system, the  $P_2$  input data rate used ranges from about 0.4 to 10MHz depending on whether a problem or a test pattern is being run. The main reason this cell was used rather than one of our 32 channel cells is that it has a much higher diffraction efficiency of about 95 percent compared to 12 percent (for the 32 channel cell) and in initial experiments we wanted as much light as possible. With the 100 ns pulse for AO2

used with test patterns, the vertical width of a data packet in AO2 is 200  $\mu\text{m}$  or half the focused optical beam size. When the finite element problem was run the pulses for AO2 were 2.3  $\mu\text{s}$  long to insure that  $P_2$  data is present for the duration of the 21  $T_1$  data packets in AO1. Thus the amount of light leaving  $P_2$  varies depending on how the data is being fed to AO2. This is not of concern since when running a given problem, we use one  $T_2$  exclusively.

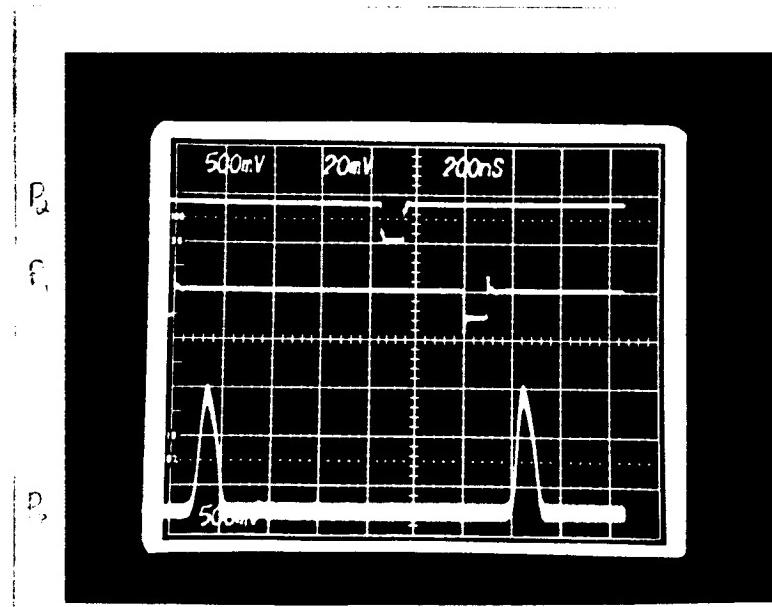
The  $P_2$  cell is driven by a driver box similar to the driver circuit we constructed for the  $P_1$  cell. It consists of a LO that drives 10 mixer-amplifier units. It has a slight offset so that a zero output is obtained with a slightly non-zero input since it was designed to be driven with TTL level signals. Since our D/A converters are not offset (i.e. they output zero volts for a zero input), the amount of input offset on the drivers reduces the useable output RF range and hence the number of levels we can represent. For the binary encoded system with one  $P_1$  channel  $M=1$ , only two output levels are needed at each  $T_1$  and the light level is sufficient for this.

We used a custom designed slit to extract only the +1 order beam from the  $P_2$  cell. Next the beam is focused onto a Merit 1900 detector by a 30 mm spherical lens. This is a very sensitive alignment since the light must not touch the barrel of the detector assembly. The problem is that if the light reflects off the inside of the barrel, any vibration of the barrel (including air currents) causes significant changes in the output. In this setup, we found that the vibrations present can easily cause the A/D to sample an invalid number. We did not cut off the barrel on the detectors since the barrel is needed to couple the optical fibers to the detector in the detector system used with the three channel system (Section 6.10).

The output of the detector-amplifier is fed into an inverting buffer. A second amplifier was necessary since the amplifier used with the detector could not be configured to invert, amplify and bias the detector at the same time. The inverting buffer was made from a Comlinear CLC-103 amplifier using a Comlinear circuit board. It has the added advantage of having a

precision offset adjustment. The output from the detector system is fed to the 100 MHz A/D converters, through the shift/add hardware and is collected by the input memory boards. This allows us to verify that we can properly collect the data from the optical system.

The next step is to set up the interface board for proper system timing. This is done by running a program in the setone directory called PP that outputs 100 ns pulses spaced 40  $\mu$ s apart to both AO cells as shown in Figure 6-18. The detector output is monitored on a scope and the timing delays are adjusted for a maximum output. The delays in the AO1 and AO2 signals are adjusted by keys on the terminal as specified in the program. The sample clock for the A/Ds is then connected to the scope along with the output of the detectors and its delay is adjusted so that the A/Ds are sampled at the proper time. These delays are then recorded and used for the various multiply routines.



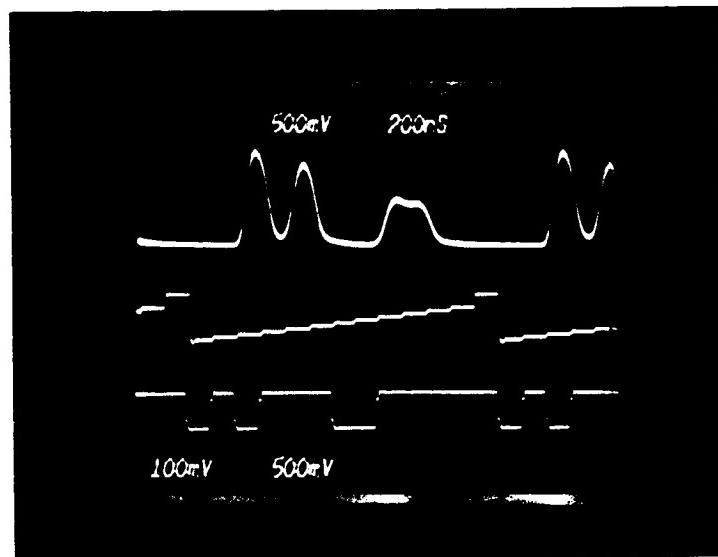
**Figure 6-18:** Detector output from the timing setup program

### 6.9.2 Results

We fed the system with various patterns and examined the output on an oscilloscope. A typical test is as shown in Figure 6-19. The bottom trace is the digital input to AO2, the middle trace is a ramp input to AO1 and the top trace is the detector output. The two input traces represent larger values as a more negative voltage (ground is the highest level on both input traces). The output trace is positive going. To time align these figures the AO2 data should be shifted to the right by two time slots (due to the delay in the AO2 cell). The three output peaks represent the proper multiplication of the third, fifth, ninth and tenth levels on the AO1 ramp by the AO2 unit pulses as shown in Figure 6-19. The detector outputs were then fed to the A/Ds to verify that the A/Ds would sample the signal correctly.

This test demonstrates the timing alignment of the system, the ability of the electronic support system to generate multi-level input data and the ability of the P<sub>3</sub> electronics to process multi-level output detector data. Our finite element case study will employ binary encoding and thus does not require multi-level inputs or outputs. In these multi-level tests, detector noise and drift were observed. The noise level of the detectors was adequate, being below one LSB of the A/Ds (12 mV). However, since the detector output tended to drift with even the slight temperature changes caused by the heating of the detector, we found it necessary to constantly re-adjust the detectors offset so that the A/D outputs was a one when required. These and similar steps were logical first steps that provided quantitative data on light budgets and tests of all system parts. These system tests proved that the optical and electronic support system was realistic.

ORIGINAL PAGE IS  
OF POOR QUALITY



**Figure 6-19:** Test outputs from the single channel test system

## 6.10 Three Channel System

This system was built to thoroughly test and exercise the entire optical and electronic support systems. We decided on three channels at  $P_2$  ( $N=3$ ) since this would allow all essential hardware to be tested and it was our first attempt at a fiber-optic faceplate. We use the same cells as in the previous system with three channels of the cell in  $P_2$  now used.

### 6.10.1 Construction

This system is the same as the system in Section 6.9 with the following exceptions:

1. The demagnification optics illuminating  $P_1$  were redesigned to reduce its sensitivity to mechanical vibration.
2. Three channels of  $P_2$  were used.
3. A SELFOC lens and fiber optic system was used to pipe the light from the  $P_3$  plane to the detector box.
4. A detector box consisting of ten Merit 1900 detectors and amplifiers was employed with three channels used.

5. The A/D converters were switched to use the third bit of the 6-bit A/Ds to determine if the output was 0 or 1. This corresponds to 000100 or a 4 and thus  $4(12) = 50$  mv of noise and drift. This was adequate for real-time operation with no adjustments for drift.

6. The functions of the output shift/add array were used.

These changes are now detailed.

The change in the demagnification optics was made after a thorough checkout of the system to determine the components most sensitive to vibration. This was necessary since even minor vibrations caused problems with the A/D output in the original system (Section 6.9). By selective testing, the demagnification setup illuminating  $P_1$  was determined to be the prime problem and the laser mount as a secondary problem source. The demagnification optics were altered by removing the pinhole and replacing the pinhole-objective assembly with a lens of the same effective focal length (10 mm) as the objective. This new lens system was sturdier and much less prone to vibration. Omitting the pinhole did not cause problems in the laser beam uniformity. The laser mounts were also tightened to minimize vibrations from that part of the system.

We then connected 3 channels of data to the second AO cell. Crosstalk was visible between channels, but it was down 20 db and did not cause a problem with our present use of binary numbers. This will become more significant when multi-level encoding is attempted.

The SELFOC assembly to couple  $P_3$  to the detector constructed consisted of a metal block holding SELFOC lenses coupled to fiber optics. The major issues to be addressed included creating a useable fiber alignment jig and a suitable adhesive to hold everything in perfect alignment when cured. Regular epoxies tend to shrink while curing causing a loss of proper alignment. The four methods we considered were:

1. a UV epoxy

2. a super glue adhesive
3. to plate and solder the SELFOCs and fiber in place
4. an RTV type adhesive

A UV epoxy cures when exposed to UV light. Problems exist with the UV epoxy method since the epoxy only cures skin deep and can thus be very fragile. We also did not have the capability to plate the fibers and SELFOCs so we tried the super glue method. We found that this type of adhesive also shrinks somewhat while curing and decided against its future use. We then used an RTV type adhesive and had reasonable results with two of the three channels and the third channel was only slightly out of alignment. It was physically moved into alignment while setting up the system. This is possible since RTV is not a very strong adhesive. The specifications of the SELFOCs and fibers used are given in Section 6.5.

This method proved satisfactory for the demonstrations needed to run the finite element problems. We later ran into misalignment problems caused by an ageing effect with the RTV. We then used a different construction method using the UV and regular epoxy that has proven to be much more stable in the long term. We first used the UV epoxy to precisely align the SELFOCs and fix them in place. We then aligned and attached the fibers with small amounts of UV epoxy and, once set, used a regular epoxy for added strength. The assembly has remained in perfect alignment for over 9 months at present.

The other end of the fibers was fed to the detector box consisting of ten Merit 1900 detectors, ten amplifiers and a precision reference used to bias the output to a suitable range. The reference used was an LM368H-10 reference connected to an LH0021CK op-amp used as an inverting buffer with a slight voltage adjustment. The output voltage was -9.00 volts. This was fed to the bias circuit on the input of the detector amplifier. The detector box also has its own power supply to reduce the number of wires and boxes used in the original detector system in Section 6.9.

We still found noise and drift problems on the detector outputs to be too severe to allow use of the least significant bit of the A/Ds. The detectors used also had a drift on the order of 50 mV/° C which is greater than the LSB (approx. 12 mV). This drift is magnified by the gain factor of the amplifiers. Since the temperature in our labs is highly variable from about 68° F to 88° F, this could cause system errors in application problems that require a long time to run. We have had good success using the top four bits of the 6-bit A/D, with the third bit of the six-bit A/D determining if the detector data is a '1' or a '0'. Airconditioning improvements would reduce room temperature variations. Our present and near term applications do not require more than 15 minutes to run. During this time, temperature drift only needs to be corrected about once a day. Our planned AC-coupled system should overcome these problems and also has a lower noise level.

This was the first real test for the shift/add array. While its operation was verified by the logic analyzer, its performance with real data at system rates from the A/Ds had yet to be confirmed. The system worked perfectly after initial minor problems were corrected.

#### **6.10.2 Results**

The system was found to work properly using various test patterns inputs to all three channels similar to those shown in Section 6.9.2. Using the low level multiply software, we found we could perform reliable 21 bit multiplies on the full system. The actual 21 bit multiplies took  $(B + N - 1)(B/N)T_1 = 23(7)T_1 = 16.1 \mu s$  to perform on the optical system. The total run time per multiply was slow since the mixed-binary to binary conversion was performed in software and the host system took significantly longer to handle the loading, unloading and conversion of all the data that was run.

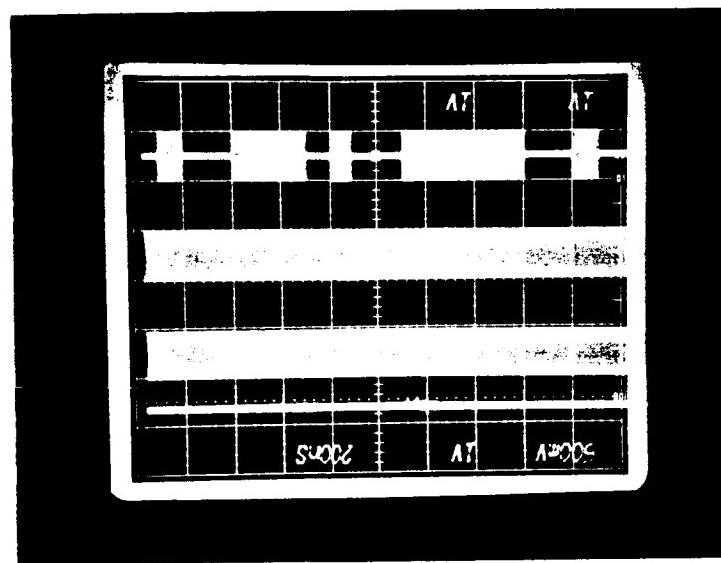
Figure 6-20 shows typical system inputs and Figure 6-21 shows the detector outputs. For the example shown, the system multiplies the  $P_1$  input 161,017 by the  $P_2$  value of 6. The lower

trace in Figure 6-20 shows the RF signal to AO1. The envelope of this signal is the 21 bit digital sequence (000100111010011111001) corresponding to the input value (161,017). Traces 3,2,1 in Figure 6-20 show the RF inputs to the three channels of AO2. These correspond to 110 respectively (the binary equivalent of the multiplicand value 6 in our example). Figure 6-21 shows the three  $P_3$  detector outputs for the  $21 T_1$  time periods. These are the products of the AO1 data and the corresponding AO2 channel values in our example in Figure 6-20. The top trace in Figure 6-21 is 0 as expected since the corresponding AO2 input is 0. The other two detector outputs are simply the AO1 data since the corresponding AO2 data is 1. This data was obtained on-line at a 10 MHz rate, thus demonstrating the performance of the system through the detector at 10 MHz.

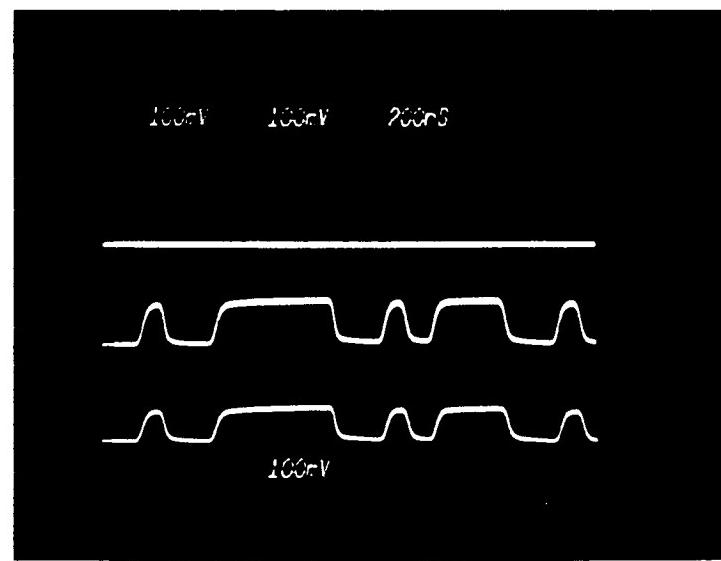
These three detector time sequential outputs in Figure 6-21 are A/D converted and fed to the shift/add network. These output data appear LSB first in time. To form the mixed-binary output, the detector 2 output is shifted left by one bit and added to the detector 1 output (which is 0 in our case). The detector 3 output is shifted left by 2 bit positions and added to the above result. Since the detector 1 output is 0, only the detector 2 and 3 outputs are of concern. The mixed-binary addition of these outputs, properly shifted, is shown in Figure 6-22. The 21 bits from detectors 2 and 3 and the 23 digits in the final mixed-binary output thus appear as shown in Figure 6-22.

Figure 6-23 shows 6 (D0-D5) of the 12 bits of the last latch at successive  $T_1$  times from left to right. The top two traces show timing signals to the system. The falling edge of the start signal initiates 2 shifts of old data present in the  $P_3$  circuitry. The falling edge of the reset pulse then initiates a sequence of B shifts and adds as discussed earlier in Section 6.4.3. For our example, the largest mixed-binary output obtained is 2. Thus, only the LSBs D0 and D1 of the last latch will have non-zero values. At successive  $T_1$  times after the reset, the D1 D0 outputs

ORIGINAL PAGE IS  
OF POOR QUALITY



**Figure 6-20:** Example RF inputs to the AO cells



**Figure 6-21:** Example detector outputs

00010011101001111001	A01 input
110	A02 inputs
<hr/>	
000000000000000000000000	Detector 1 output
00010011101001111001	Detector 2 output
00010011101001111001	Detector 3 output
<hr/>	
0001101221101222210110	Shift/Add output
<hr/>	
00011101011110111010110	Final binary output

**Figure 6-22:** Action performed by the shift/add on the example problem

(left to right in Figure 6-23) are : 00,01,01,01,00,01,10,etc. These correspond to the mixed-binary output: 0,1,1,0,1,2,etc. These correspond to the 6 least significant digits in the mixed-binary output in our example. The remaining outputs in Figure 6-23 correspond to the remaining digits in the result. The mixed-binary output ends after 23 time slots at which point the reset pulse reappears in Figure 6-23.

Figure 6-24 shows the converted binary representation of this mixed-binary output. this is obtained in software on our system. The standard binary output for our example is : 011101011110111010110 as shown in Figure 6-24.

## 6.11 Summary

In this chapter we described and demonstrated an optical matrix-vector computer architecture and a very flexible electronic support system. The system built can perform 21-bit multiplies in  $16.1 \mu s$  and can be expanded to perform 30-bit multiplies in under  $0.5 \mu s$ . The system can also be expanded to compute a 10 element VIP in  $0.5 \mu s$  (10 multiplications and additions) or one multiplication/addition every 50 ns.

The electronic support hardware consisted of a host computer, a high speed memory subsystem, A/D and D/A conversion hardware and custom shift/add circuitry. The host computer is a Intel 286/380 system that was customized for this system by adding a high speed

ORIGINAL PAGE IS  
OF POOR QUALITY.

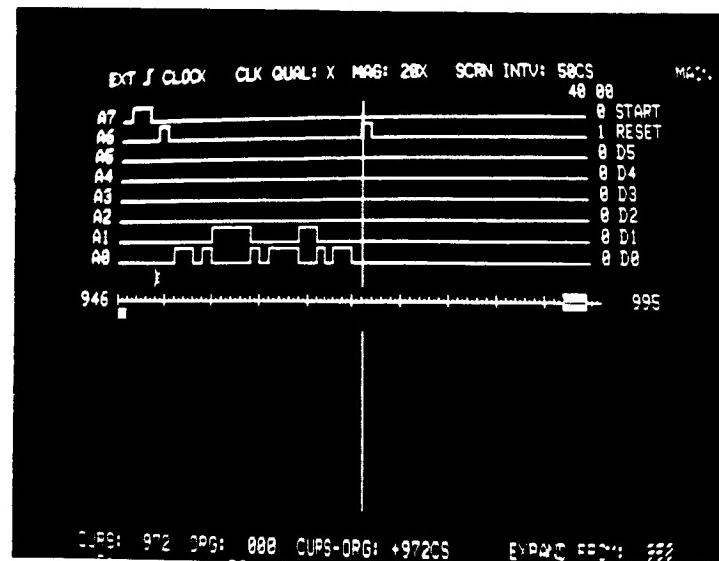


Figure 6-23: Example mixed binary outputs

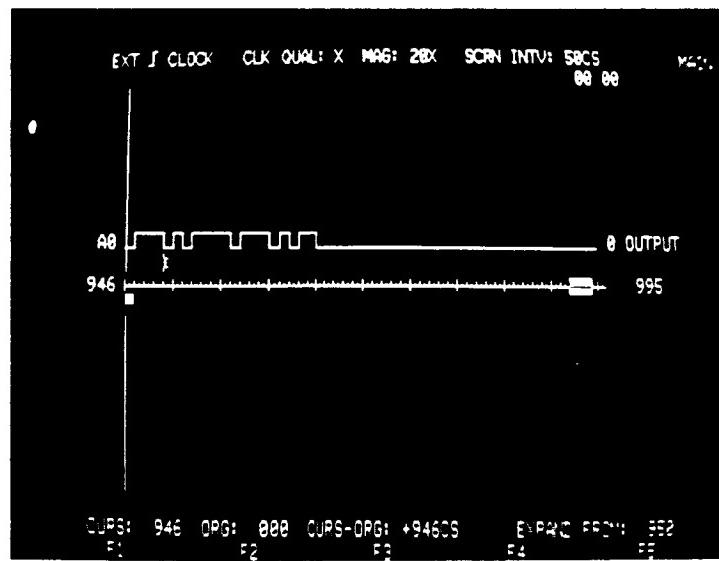


Figure 6-24: Example system final output

math co-processor, extra I/O ports for communication with the CEODPs VAX 11/750, a 1/2" tape drive and special software to run the optical system software. The software written included routines for diagnostics, calibration and operation of the digital and optical system. This includes multiply routines that allow system users to interface easily to the optical processor and an LAE solution by the LU decomposition method. The four high-speed memory boards feeding the system are capable of supplying 32 12-bit channels at a 10 MHz rate with each channel holding 4096 words of information. The two input memory boards have the capability of accepting 16 channels by 12-bits of data at similar rates. We have demonstrated working hardware that includes 4-bit 200 MHz D/As and 6-bit 100 MHz A/Ds. We also built and tested an ECL shift/add output array that emulates a CCD detector at 6-bits and 10 MHz speeds per channel (i.e. a 9-bit 10 MHz CCD array) The shift/add card was designed to work at 40 MHz. Since this was the first attempt at much of this hardware, it has resulted in working designs for use with larger systems with faster electronic support.

We fabricated and tested the electronic support and optical processor and obtained quantitative data for light budgets and noise levels. We demonstrated that we could operate our test system at a 10 MHz clock rate with no problems and can foresee no problems with clock rates of up to 40 MHz, which is the highest speed we presently anticipate using.

The major purpose of this part of the project was to assemble the electronic support system to allow us to obtain quantitative data on an initial optical lab matrix-vector test bed and to define and qualify directions for future research on such systems. The highlights of our work are now Itemized:

- The electronic system requirements for an optical matrix-vector processor with M processor channels with N digits accuracy and multi-level encoding was quantified. (Section 6.2)
- A new electronic support system with an Intel host processor and superior hardware and software support was designed and fabricated. (Section 6.3)

- The hardware system provides much higher data rates and accuracy than any other previous optical matrix-vector system.
- A new technique to realize any desired accuracy using any number of digits on the processor was devised and demonstrated. (Section 6.7.1)
- The software routines for the system are much more user friendly than with any other optical matrix-vector system. (Section 6.6)
- The first experimental demonstration of a direct LAE solution on an optical processor was provided. (Section 6.10)
- The ability of the electronic support system to handle multi-level data was demonstrated. (Section 6.9.2)
- The optical and electronic support was shown to produce practical 21-bit accuracy multiplications with an error rate below  $10^{-7}$ .
- We obtained quantitative data on the light budget, noise and drift that will prove valuable in building faster versions of this processor.

## 7. LABORATORY OLAP PERFORMANCE AND PLANS

The prototype laboratory OLAP was described earlier in the Spring 1986 report to NASA<sup>31</sup>. Some qualitative and quantitative results of initial performance tests for single multiplies and multiplication tables were reported earlier. Since that time, the laboratory OLAP system was used to run a static finite element plate bending case study. The case study was detailed elsewhere. The results of this OLAP demonstration are given in Chapter 5 of this report. In this present chapter, we discuss the current OLAP performance limitations, and our plans to decrease or eliminate them with a new AC-coupled operating mode.

### 7.1 Laboratory OLAP Characterization

The laboratory OLAP system was extensively described in Chapter 8 of our Spring report to NASA, and in Chapters 5 and 6 of this report. Only a brief description of the system is given here for reference purposes.

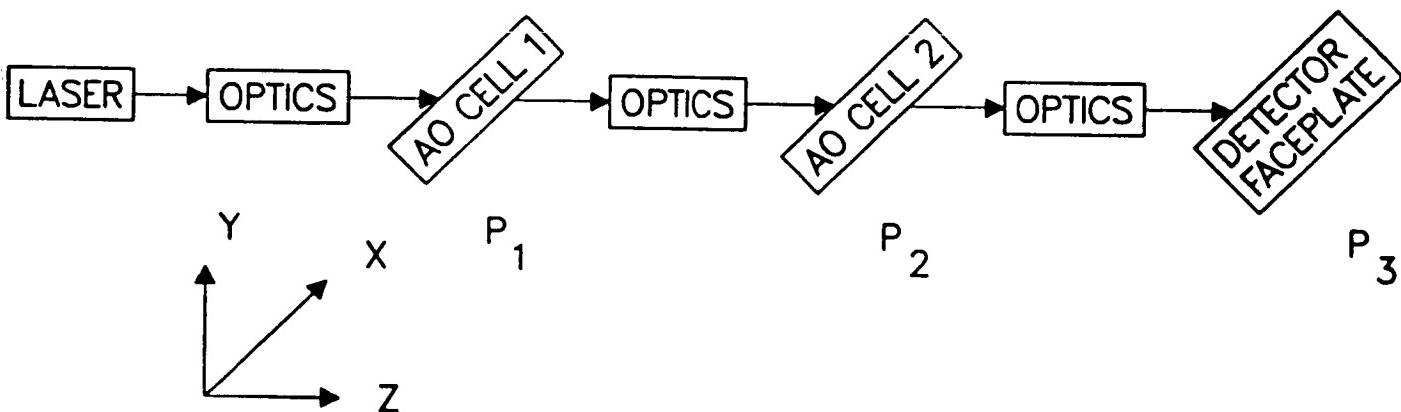
#### 7.1.1 Laboratory OLAP System Review

A basic schematic of the laboratory optical system used is shown in Figure 7-1. The blocks at  $P_1$ ,  $P_2$ , and  $P_3$  are tilted for illustrative purposes only; the actual component orientations are better illustrated in other figures. The laser beam is first compressed (demagnified) by a combination of lenses in order to properly illuminate the AO cell at  $P_1$ . The outputs of the  $P_1$  AO cell 1 are the zero-order and the first-order modulated beams. The zero-order beam is blocked by a spatial filter, and another combination of lenses shapes the first-order beam such that it properly illuminates the AO cell 2 at  $P_2$ . The zero-order output of the  $P_2$  AO cell is blocked by a spatial filter, and the first-order is imaged onto a  $P_3$  fiber-optic detector faceplate.

There are  $M$  channels at  $P_1$  and  $N$  channels at  $P_2$ . The  $N$  bits of the  $M$  multipliers are fed bit serially into the  $P_1$  channels, and the  $N$  bits of the multiplicands are fed in word parallel

form to the  $P_2$  channels. The convolutions of the multiplier and multiplicand bit streams are summed onto the detector plane and appear in mixed radix form. The proper shift and adds take place in ECL hardware to form the desired products.

In the current laboratory OLAP,  $M=1$  and  $N=3$ . By using partial product partitioning of the digital multiplication by analog convolution algorithm, we perform 21-bit multiplies. By using seven partitions and with  $N=3$ , we obtain a 21-bit processor.

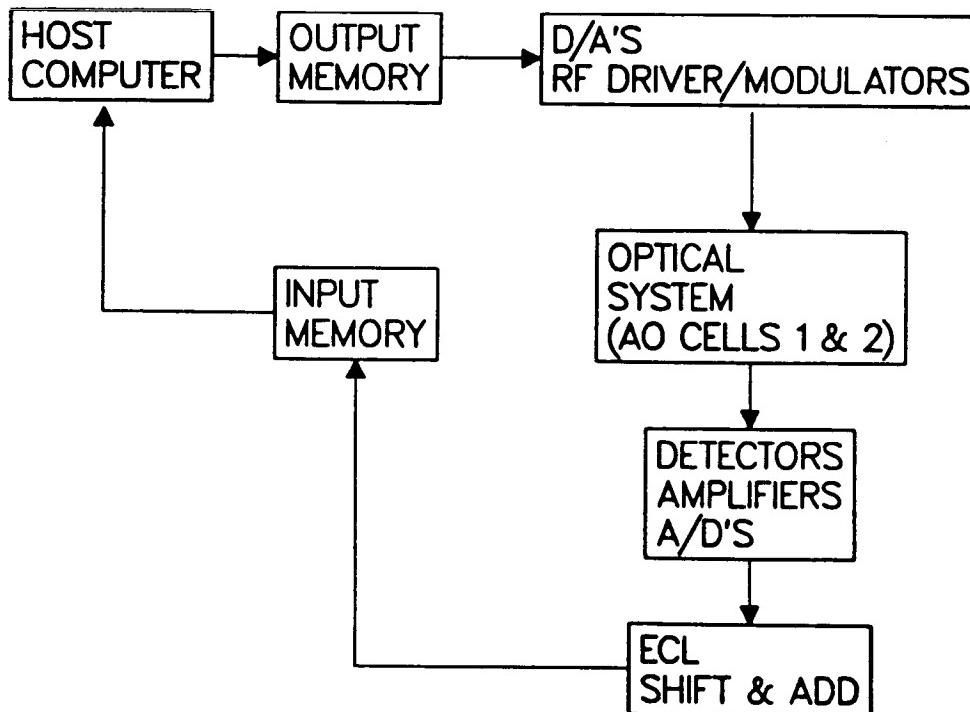


**Figure 7-1:** Laboratory Optical System Schematic

A block diagram of the laboratory system is shown in Figure 7-2. This diagram emphasizes and illustrates the role of the electronic support hardware components. The support hardware can be divided into four system components: the computer input and output high-speed memories, the digital-to-analog converters (D/As) and RF driver/modulators for the AO cells, the detectors/amplifiers and analog-to-digital converters (A/Ds), and the emitter-coupled logic (ECL) shift and add detector hardware.

The high-speed memories are run at 10 MHz. The OLAP is tested in a burst processing mode, where the output memories are loaded with the required data and dumped to the OLAP at 10 MHz. One 12-bit output memory channel from the host computer is used to feed each of the  $P_1$  and  $P_2$  input AO channels. The D/As are four-bit converters which feed the required

levels to the RF driver/modulators for the  $P_1$  and  $P_2$  AO cells at 10 MHz. At  $P_3$ , a fiber-optic faceplate collects the output light from  $P_2$  and routes it to the detector/amplifier box. The detected optical signals are amplified to the proper levels and then sent to six-bit A/Ds at 10 MHz on each output detector. The digital outputs are then processed by the ECL shift and add hardware system. The input memories collect data from the ECL shift and add hardware at 10 MHz.



**Figure 7-2:** Laboratory System Block Diagram

### 7.1.2 System Performance Limitations

Chapter 5 of this report describes how the laboratory OLAP performed very well when running the static finite element plate bending case study. The processor ran with  $M=1$ ,  $N=3$ , and  $B=2$ , ie. binary encoding. Earlier,<sup>25</sup> we discussed the digital error source simulation of the OLAP for this same case study. Our laboratory results agree with the digital simulation results by showing that the optical error sources are at a low enough level to allow error-free processing with binary encoding. The specific error source levels were documented previously<sup>31</sup>. No statistical error rate was rigorously determined, but from the laboratory operation, it can be estimated at lower than 1 bit error in every  $10^7$  bit multiplications. This estimate was obtained by continuously running batch jobs on the system which performed all sizes of 21-bit multiplies on the optical processor.

Although the laboratory OLAP works quite well in its present configuration, some limitations do exist. These limitations were determined in the initial tests of the laboratory OLAP setup, and were noted earlier<sup>31</sup>. No new significant limitations were discovered when the laboratory OLAP was tested for the initial case study. The two major limitations are light level and detector drift. Both of these factors must be improved if we are to expand the laboratory OLAP (in terms of the number of channels  $M$  and  $N$ , or in terms of the base  $B$  that we use).

The light levels at  $P_3$  of the OLAP are sufficient for operation with  $N=3$  and  $M=1$ , but an increase in  $N$  primarily (and  $M$  to a lesser extent), would decrease the light available at  $P_3$ . If the number of channels  $N$  were doubled, the light levels at  $P_3$  would be reduced at least by a factor of 2. The decrease in light level is not as severe if  $M$  is increased, and mainly depends on what type of modulator is used at  $P_1$ . If we want to increase the base  $B$  used, more dynamic range at the detectors, and thus a larger light level is required. Our detector/amplifier output (for a binary 1) is currently about 50 mV, the detector/amplifier noise is approximately 10 mV

peak to peak, and the A/D step size is 12.5 mV. Thus it is obvious that more output range is needed if B is to be increased.

Currently, the light incident on a single channel of the  $P_3$  fiber optic faceplate is approximately 70  $\mu$ Watts, representing a binary 1. With a 3 dB loss through the optical fiber coupling, approximately 35  $\mu$ Watts are incident on the solid state detectors. The light entering the optical system at  $P_1$  is about 20 mWatts. Thus, the light loss thru the system for  $M=1$  and  $N=3$  is approximately 30 dB. The shot noise floor of the detectors is several  $\mu$ Watts, thus the  $P_3$  light levels cannot be lowered significantly. To remain at similar  $P_3$  light levels while increasing N or M, the amount of light into  $P_1$  needs to be increased. To increase B, we need more light at  $P_3$  to yield more dynamic range, thus we also need to increase the amount of light into  $P_1$ . This could be accomplished by using a more powerful laser, or tuning up the one we have (it is capable of 50 mWatt operation). However, a more light-efficient scheme is to use a high-power (20 to 30 mWatts) laser diode at each  $P_1$  channel. Thus, as M is increased, we increase the amount of optical power input to the system. More important, we do not use an acoustooptic modulator at  $P_1$ , which currently only has a diffraction efficiency of only about 5%.

The second important OLAP limitation is detector drift. The drift is due to thermal effects and has two sources. The first, and probably the most important, is the ambient temperature instability in our laboratory. The second source is the heating (or relative cooling) of the detectors when they encounter a number of consecutive binary 1's (0's) during processing. The detector drift is often significant, as the detectors are specified to drift 10 mV/degree C, which is about 20% of our full scale output. The laboratory temperature instability is correctable, and University plans call for correction of the problem, but it is not clear how soon the problem will be corrected. The second thermal effect source is not as easy to eliminate. The temperature of the detectors will always vary with the DC level of the incident light, which is

solely dependent on the numbers being processed. One solution that would eliminate the drift problem from both sources is to AC couple the entire system. This approach will be detailed in the next section.

At this point, another critical aspect of the laboratory OLAP merits attention, and that is timing. As N and M increase, the system timing becomes much more difficult. This involves ensuring that the data at  $P_1$  and  $P_2$  are in the right place at the right time, and that the A/D samples and processes the detector/amplifier outputs at precisely the correct time. As the number of channels M and N in the system increase, more degrees of freedom are introduced which must be properly synchronized. Thus, as we increase M and N, we expect considerably more precision to be required in system timing.

## 7.2 AC-Coupled OLAP

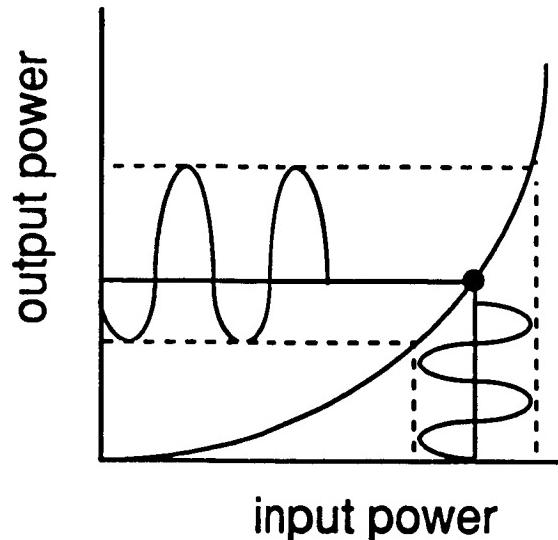
The previous section discussed the laboratory OLAP system problems that limit the increase in the number of  $P_1$  channels M, the number of  $P_2$  channels N, the speed, and the encoding radix B. We have concluded that increases in M,N, and B will need to be accompanied by an increase in the light level through the processor, and the elimination of detector drift. We plan to increase the light level by using laser diodes for the  $P_1$  point modulators, as discussed above. The problem of the detector drift remains, and the proposed solution is to AC couple the system. This will negate the detector drift effects, which are essentially time-varying DC components. The AC coupling is also necessary to properly operate the laser diodes without substantial drift in their optical power output.

### 7.2.1 AC Coupling Basics

To AC couple the optical system, the input light (from the laser diodes) will be AC modulated. In the present system, the  $P_1$  point modulator (acoustooptic cell) passes light to represent a binary 1, and does not pass the laser light to represent a binary 0. These pulses of light and no light occur presently at 10 MHz, the data rate for the memory system. In the AC coupled system, the zero level for the light will be some fixed intensity level, since we cannot talk about negative light intensities. A zero from  $P_1$  will thus be light at that fixed level, i.e. a signal with an AC component of zero. To produce a binary 1 (or some other level if  $B > 2$ ), the light will be amplitude modulated on a 300 MHz sine wave carrier input to the laser diode about the zero level. The amplitude of the sine wave will determine the value of the bit. The  $P_2$  modulator will be a multi-channel acoustooptic cell as before, since the product of an AC signal ( $P_1$ ) and a DC signal ( $P_2$ ) is an AC signal. The detector/amplifier output will be AC coupled and amplitude demodulated before being sent to the A/Ds. The A/Ds will thus see a signal that will be uncorrupted by the slow detector drifts, since they are at or near DC and are not passed through the AC coupled system.

### 7.2.2 Laser Diode Modulation

Laser diodes operate at a constant optical power output when a DC driving voltage is applied. A typical laser diode operating curve is shown in Figure 7-3. The laser diode operating curve is unfortunately fairly sensitive to temperature variations, and the laser diode will heat up at higher operating points. Thus, it is difficult to operate at discrete points on the operating curve and avoid transient drift affects. These are due to the temperature of the laser diode changing between operation points, depending on how much time is spent at each operating point. If the laser diode is AC amplitude modulated around an operating point, as shown in Figure 7-3, the temperature of the laser diode and thus its output power will remain constant. This is how the laser diodes will be driven in our AC coupled system. The DC operation point on the curve represents the zero level discussed above.



**Figure 7-3: Laser Diode Operation Curve**

In order to effectively demodulate an amplitude modulated signal, the carrier frequency must be substantially greater than the modulation frequency. The modulation frequency is simply the data rate of the optical system, which is 10 MHz. As discussed earlier,<sup>31</sup> we plan to increase the system data rate to 40 MHz, by obtaining high-speed ALUs and by multiplexing and demultiplexing the high-speed memories. This data rate conversion depends on cost and availability of the ALUs, and thus we are not sure when this change is realistic. However, we will describe and build the AC coupled system to be able to handle 40 MHz modulation. We will use a carrier frequency of 300 MHz, which is more than seven times the highest modulation rate frequency (40 MHz) planned. Thus, the frequency spectrum of the light leaving  $P_1$  will have a component at DC, and a non-zero band between 260 MHz and 340 MHz (for a 40 MHz data rate, double side band).

### 7.2.3 Laser Diode Imaging Optics

When using laser diodes for the  $P_1$  point modulators instead of an acoustooptic cell, different imaging optics between  $P_1$  and  $P_2$  will be required. There are two things which make the light distribution from a laser diode more difficult to control than that from an acoustooptic cell. First, the physical size of each laser diode is much larger than the width of an acoustooptic channel. The width of an acoustooptic channel is typically 1 mm, and they are usually spaced a few mm apart in a multichannel cell. The package size of a laser diode is presently typically 1 cm for discrete laser diodes, requiring the center spacing between laser diodes to be at least 1 cm. Since the distance between the M channels at  $P_2$  is a few mm's, it is much harder to demagnify a laser diode array output light distribution than that from a multi-channel acoustooptic cell. The second problem with laser diodes is that their output beam is not a thin collimated beam, as with a gas laser. The output light is in the form of a rapidly diverging elliptical beam, although it is highly coherent. The rapidly diverging beam requires the use of low f number optics, and the elliptical shape means that different optics will be needed for the major and minor axes.

In commercial applications (laser printers, CD players, etc.), the laser diode light is harnessed by a small tube containing multiple lens elements, which fits over the laser diode package. This device is known as a collimating pen, and it produces a collimated beam output with low divergence. Unfortunately, the price of collimating pens is still quite high (\$500-\$1000 each), and most are custom made for specific laser diodes. Thus, we will be using regular laboratory optics to handle the laser diode light, and this will require a considerable amount of effort. We plan to use laser diodes at a wavelength of  $\lambda=780$  nm, and the peak responsitivity of the detector array is appropriately near this wavelength. Some new low F-number optics will need to be purchased, along with a machined plate to hold the laser diodes. This equipment will cost approximately \$1000.00. One novel approach for collimating an array of laser diodes is to use a computer generated hologram (CGH). However, we feel this effort would be too extensive

by itself to warrant planning to use it in our system. There is also an issue of the light transmittance efficiency of a CGH.

#### **7.2.4 AC Coupled Detector System**

With the light in the OLAP modulated on a 300 MHz carrier, the detectors must be able to respond to light at that temporal frequency. The detectors that are being used in the current OLAP only respond to frequencies up to about 100 MHz, thus, a different detector system is required. United Detector Technology manufactures a detector array that is suitable for our needs. It is a 10-element silicon array with a spacing of 1.65 mm between detector elements. The light distribution from  $P_2$  will be imaged directly onto the detector array, eliminating the need for a fiber-optic faceplate to guide the light to the detectors. A fiber-optic faceplate is desireable, particularly with discrete detectors, and when control of the spacing of the detector plane (fiber-optic) inputs is desired. However, fabricating and aligning a fiber-optic faceplate is a difficult and time-consuming process, and there is always a loss in optical power due to coupling losses of about 3 dB. The detector array that we will use has a very wideband frequency response into the GHz range, thus it will operate well around 300 MHz.

The detector array elements are silicon PIN diodes. These detectors require a -10 V reverse bias to operate. To amplify the current generated by the detectors, a wideband amplifier is needed, and a low input impedance transimpedance amplifier is typically used to preserve linearity. We have made arrangements with General Fiber Optics Inc. to provide us with an amplifier system that will also house the detector array. The amplifiers will respond to the 260 MHz to 340 Mhz bandwidth, and will produce a transimpedance gain of approximately  $10^6$ . The cost of the unit will be approximately \$4600.00.

The output of the detector amplifiers will feed an amplitude demodulation circuit. The signal will first enter a high pass filter to remove its DC component. The output of the high

pass filter will be input to an envelope detector circuit made up of an RF diode bridge and capacitors. The output of the envelope detector will pass through a 150 MHz low pass filter to smooth out the ripple. This demodulated signal will then be sent through a bias-T to provide the proper offset for input to the A/D converters. Each detector demodulator unit will cost approximately \$90.00.

### 7.3 Future Plans

The plans for the immediate future are exactly those that have been outlined in the previous sections. Our goal is to increase the channel capacity of the laboratory OLAP ( $M$  and  $N$ ), and to use a radix  $B$  larger than 2. We have described the steps we feel must be taken to achieve this goal.

We have the laser diodes and driver circuits in our labs already. The cost of the laser diodes is about \$250.00 each, and the driver circuits are approximately \$50.00 each. We are in the process of selecting the imaging optics we will use. The detector arrays have arrived, and one (plus a spare) has been sent to General Fiber Optics for placement into the detector amplifier unit. This unit should be delivered to us by March 1, 1987. We have prototyped the demodulator circuits in our lab, and we are currently testing various diodes and filters.

Once we have all the hardware together, we will proceed with the new laboratory OLAP. Initially we will prototype a single channel system, i.e.  $M=1$  and  $N=1$ . This will let us evaluate our basic AC-coupled design and it will provide insight into the new engineering issues we must consider. Once we have satisfactorily finished with this single channel prototype, we will continue with the multi-channel expansion. We will first increase  $N$ , probably to 5 channels, and have an operational OLAP. We can then consider increasing  $N$  to 10 channels, and then increasing  $M$ . We will start with  $M=2$ , and continue to  $M=3$  or more. We will then increase  $B$ , using  $B=4$  since it is a power of two. Data obtained on the laser diode imaging optics and the light budget will be most useful.

## 8. CASE STUDIES FOR SIMULATION AND TESTNG OF THE OPTICAL LINEAR ALGEBRA PROCESSOR

### 8.1 Introduction

We plan to address the finite element and finite difference solution of two separate problems from computational fluid dynamics (CFD) and one from structural dynamics. Each study will first be implemented in software that simulates the data flow and error sources of the optical processor and then on the laboratory optical processor. The studies are modest in size due to the large amount of computer time needed to simulate the operation of an optical processor. Each of the case studies will be executed with the simulation software on a Cray X-MP/48 operating out of the Pittsburgh Computing Center, in Pittsburgh, Pa. Our choice of algorithms depends upon several factors: how the algorithms direct data flow through the optical processor, the amount of accuracy we require in the final solution, the computation time of the algorithms, and how each algorithm is affected by errors that are particular to the optical processor.

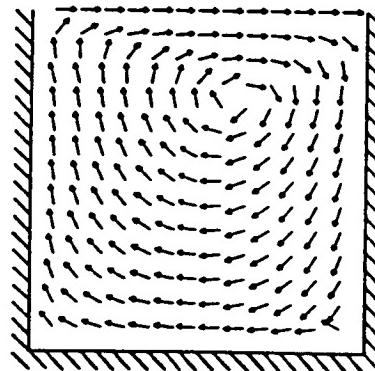
### 8.2 Computational Fluid Dynamics

The two chosen CFD case studies invoke both steady-state and transient motions of nonlinear fluid motion in a cavity domain with finite element and finite difference formulations. Each case study will be implemented in two stages. First, each will be executed in a software simulation of the optical processor in which data flow and error sources of the processor are simulated. This implementation will predict the optical processor's performance in laboratory operation. The results will be quantified numerically and displayed with appropriate graphical tools. Second, each finite element/difference study will be implemented on the laboratory optical processor.

The CFD studies are formulated with the methods of finite elements and finite differences. Both methods produce a system of algebraic equations which are readily implemented on our optical linear algebra processor. The finite element and finite difference discretizations will be performed externally to the OLAP; the resulting algebraic equations will be solved through matrix-vector operations on the optical processor. We now detail the two CFD studies.

### 8.2.1 Nonlinear, Steady-State CFD

The first case study formulates the 2-dimensional Navier-Stokes equations over a rectangular region  $\Omega$  by the method of finite elements. It will be implemented first in a software simulation of the optical processor and then on the laboratory optical processor. The Navier-Stokes equations are well-known in fluid mechanics and describe either time-varying or steady-state incompressible viscous flows and are highly nonlinear. An example is fluid motion in a driven cavity; i.e., a 2-D slice of a rectangular domain containing incompressible viscous fluid where one surface is set in motion, while in contact with the fluid, thus creating fluid motion within the cavity. A 2-dimensional velocity vector diagram depicting what the fluid motion in such a cavity might look like is shown in Fig. 8-1, where the moving surface is the top of the cavity.



**Figure 8-1:** Flow in a Driven Cavity

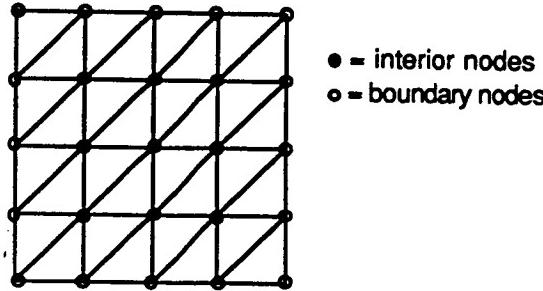
In our CFD study we seek a finite element solution to the 2-D steady state Navier-Stokes equations,

$$(\mathbf{u} \bullet \nabla) \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} = \mathbf{f} \quad \text{in } \Omega \quad (8.1)$$

$$\nabla \bullet \mathbf{u} = g \quad \text{in } \Omega \quad (8.2)$$

$$\mathbf{u} = 0 \quad \text{on the boundaries of } \Omega \quad (8.3)$$

where  $\nu$  is the coefficient of viscosity which describes how much drag the fluid will incur when set into motion,  $p$  is pressure,  $\mathbf{u}$  is a two-component velocity vector,  $\mathbf{f}$  is a 2-component force vector due to external forces applied to the system,  $g$  is a nonzero function chosen specifically to allow for an exact solution of the Navier-Stokes equations, and  $\bullet$  indicates dot product. The region  $\Omega$  is rectangular and discretized by the finite element method into triangular elements as shown in Fig. 8-2. The unknowns we seek are two velocity components at each finite element grid node and a pressure value within each finite element. The nonzero right-hand side of (8.2) indicates that mass is being created in  $\Omega$  with distribution  $g$ . Unlike a *driven cavity* flow, where one cavity boundary is set in motion, our case study has all boundary velocities set to zero. It is the function  $g$ , i.e., the mass distribution, which induces fluid flow within the cavity.



**Figure 8-2:** Discretized Driven Cavity Domain

We have obtained a finite element program with the above problem description from Dr. Janet Peterson of the University of Pittsburgh<sup>32</sup>. We will alter this program so that it simulates the data flow and error sources of our optical processor. The program user may vary the number of nodes in the finite element mesh, thus helping to reduce the program's CPU time by choosing a small number of grid nodes. The minimum number of nodes we will use is 5 on a side of  $\Omega$ , or 32 triangular elements and 25 nodes for our square  $\Omega$ . Since 16 boundary node values

are known from the boundary conditions, only the velocity components at each of the 9 interior nodes are unknown. There is an additional unknown, i.e., pressure, within each of the 16 mesh boxes (two triangular finite elements per box). Thus, there are a total of 2x9 velocities plus 16 pressures or 34 unknowns. The resulting finite element matrix equation takes the general nonlinear form,

$$[K(u)]u + c = 0, \quad (8.4)$$

where uppercase letters denote matrices and lowercase letters denote column vectors. The vector  $c$  contains known forces and parameters. We seek the unknown vector  $u$  which includes velocities and pressures. Its length, for the above case, is 34 floating point elements. Thus,  $K(u)$  is a (34 x 34) sparse matrix.

Since the equations are nonlinear, ( $K(u)$  depends on  $u$ ), the Newton-Raphson method will be used. Since Newton-Raphson is an iterative algorithm, a good initial guess for the solution vector  $u$  is helpful. Under usual circumstances, a researcher will have some idea of the general behavior of the system under study, hence, obtaining a reasonable initial guess to the solution vector is realistic. In our case study, the finite element program supplies the exact solution to the Navier-Stokes equations so we can choose an initial  $u^{(0)}$  that will ensure convergence of the Newton-Raphson algorithm. (Even though most experimental studies do not have exact solutions on which to base an initial guess, the engineer will have a physical understanding of the problem under study and can thus make a reasonable initial guess to the solution).

To illustrate the Newton-Raphson method we let  $\Psi(u)$  be the vector function given in (8.4); i.e.,

$$\Psi(u) = [K(u)]u + c = 0. \quad (8.5)$$

Each iteration step  $k$  in the Newton-Raphson method produces a system of linear algebraic equations

$$J^{(k)}(u^{(k+1)} - u^{(k)}) = -\Psi(u^{(k)}), \quad (8.6)$$

which is of the form  $\mathbf{Ax} = \mathbf{b}$  in matrix notation, and  $J^{(k)} = \Psi_{\mathbf{u}}(\mathbf{u}^{(k)})$  is the Jacobian matrix whose elements are  $\Psi_{\mathbf{u}}$  (the partial derivatives of  $\Psi$  with respect to the vector  $\mathbf{u}$ ). We can solve these linear equations by direct or indirect linear equation solver algorithms. We propose using LU decomposition to solve the linear equations of (8.6) at each iteration step and a difference approximation to the partial derivatives of the Jacobian matrix.

The Newton-Raphson procedure is as follows:

1. Choose an initial  $\mathbf{u}^{(0)} \equiv \mathbf{u}^{(k)}$ .
2. Calculate the Jacobian matrix for  $\mathbf{u}^{(k)}$ . Calculate  $\Psi(\mathbf{u}^{(k)})$  using  $\mathbf{u}^{(k)}$  and equation (8.5).
3. Insert the results from step 2 into equation (8.6) and calculate the new  $\mathbf{u}^{(k+1)}$ . This will be done by solving (8.6) for  $\mathbf{u}^{k+1} - \mathbf{u}^k$  by LU decomposition of  $J^k$  and subsequently adding  $\mathbf{u}^{(k)}$  to the resulting  $\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$ .
4. Insert  $\mathbf{u}^{(k+1)}$  from step 3 for  $\mathbf{u}$  in (8.5), and determine if it is an acceptable solution to (8.5), i.e., determine if  $\Psi(\mathbf{u}) < |\epsilon|$ , where  $\epsilon$  is an acceptable error range for a solution to (8.5). If the error condition is met, then stop, and  $\mathbf{u}^{(k+1)}$  is an acceptable solution vector to (8.5). If not, return to step 2 using  $\mathbf{u}^{(k+1)}$  as the  $\mathbf{u}^{(k)}$  in (8.6) and repeat steps 2,3 and 4. Continue until a solution vector is found.

The computational burden of calculating a new Jacobian matrix at every iteration step can be reduced if we calculate  $J^{(k)}$  only occasionally. We will use the Newton-Raphson procedure with the Jacobian from the *first* iteration for all subsequent iteration steps. The process will not converge as quickly, but it does not require the calculation of the Jacobian at every iteration step. Alternatively, we will investigate calculation of the Jacobian once every  $m$  iteration steps. This will speed-up convergence but will add to the total computation time of the algorithm.

To evaluate the effects of iterative and direct methods of solution on the optical processor, we will, for our case study, also use an iterative method (Gauss-Seidel or SOR) to solve the equations in (8.6). Comparisons will be made between the iterative and direct methods (LU decomposition) and their effects with respect to OLAP performance and error sources.

### 8.2.2 Nonlinear, Transient CFD

In the second CFD case study, we seek a finite difference solution of a nonlinear, transient flow within a driven cavity. Unlike the previous case study, mass is not injected into the cavity domain. The fluid is set in motion by contact with a moving boundary as in Fig. 8-1. The governing equations are the conservative stream-function and vorticity form of the 2-dimensional incompressible Navier-Stokes equations, and transient motion of the flow will be included. A computer program with this problem description has been written and obtained from Dr. Robert E. Smith, NASA Langley Research Center<sup>33</sup>.

The normalized stream-function and vorticity conservation form of the 2-dimensional incompressible Navier-Stokes equations are:

$$\zeta_t = -(\psi_y)\zeta_x + (\psi_x)\zeta_y + (\zeta_{xx} + \zeta_{yy})/R , \quad (8.7)$$

$$\psi_{xx} + \psi_{yy} = -\zeta , \quad (8.8)$$

where  $\psi=\psi(x,y)$  is the stream function,  $\zeta=\zeta(x,y)$  represents vorticity and R is the Reynolds number of the fluid. The stream function and vorticity are both scalars in 2-dimensional problems. Therefore, with a 1/16 grid size; i.e., a rectangular region with 16 boundary nodes and 9 interior nodes, there will be  $2 \times 9 = 18$  unknowns. Thus, the solution vector will be comprised of 18 floating point elements.

We will consider the alternating-direction-implicit (ADI) method to solve the vorticity equation (8.7) and the successive overrelaxation (SOR) method to solve the stream function equation (8.8). As in the first case study, this program will be altered to emulate the data flow and error sources of the optical processor. The execution of this program will simulate the solution procedure of the optical processor. The program has the capability to vary time step, grid size, Reynolds number and initial conditions. After implementing this case study in the simulation software, the case study will subsequently be implemented on the laboratory set-up of the optical processor. Comparison between the actual laboratory performance and the predicted

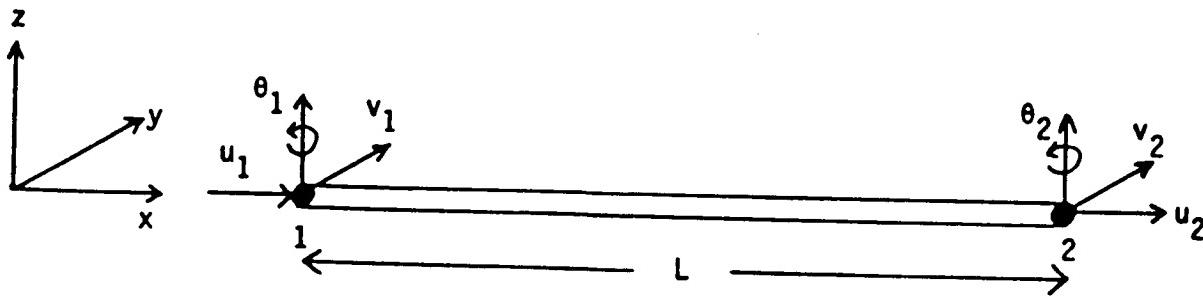
behavior by the simulation software will be used to upgrade the simulation software so that it is more representative of the actual laboratory performance. The availability of such a program will allow us to more quickly predict the effects of architectural or algorithmic changes that we may choose to make in the optical processor or the effect of optical processor errors on algorithmic changes.

### **8.3 CFD Summary**

The implementation of each of the CFD case studies will consist of two stages. The first will be carried out in software on a digital computer. This requires the creation of software that simulates the data flow and possible error sources of the optical processor. Each study will be executed with this software to predict the laboratory performance of the optical processor. The second stage will be the implementation of each finite element and finite difference problem on the laboratory optical processor. The laboratory tests will allow us to investigate possible improvements to the processor. Comparison of the predicted behavior (via the simulation software) and the actual laboratory performance will allow insight into the validity of our simulator error models, and the performance of the optical processor. Such comparisons will allow us to improve the simulation program so that it may more accurately simulate the behavior of the actual processor so that it can be used to more quickly evaluate the effects of optical processor architectural changes or algorithmic changes. The simulation process is expected to be time consuming, in both man-hours and CPU time, as demonstrated by previous experience with a linear static structural mechanics case study which required over three hours of CPU time to execute. The CFD studies are more complex since both are nonlinear and one includes transient motion. The Cray X-MP/48 at the Pittsburgh Supercomputing Center will be used for these tasks.

## 8.4 Linear Dynamic Structural Mechanics Case Study

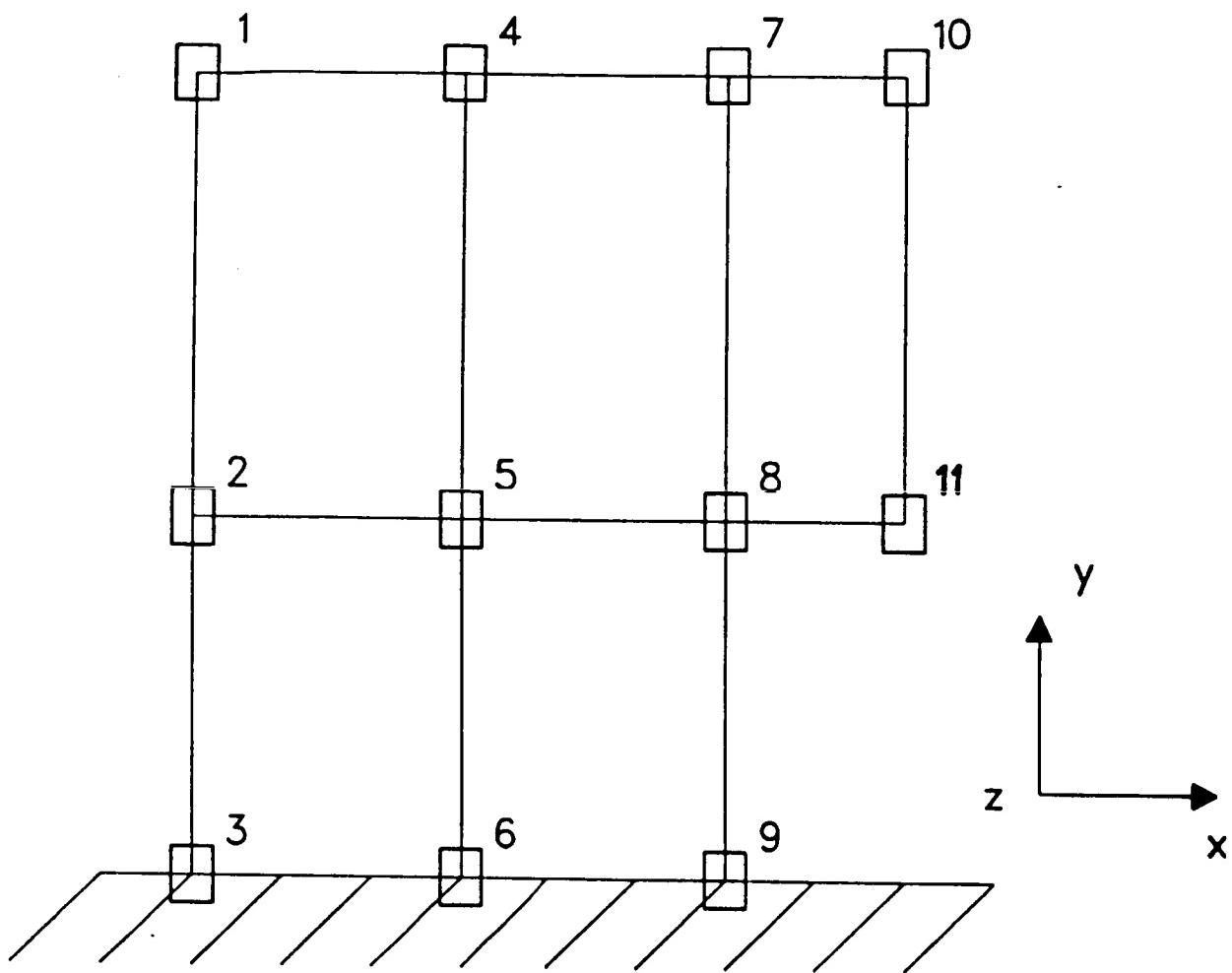
The second structural mechanics case study is a linear dynamic finite element problem. It is a plane frame analysis problem of a structure composed of standard beam elements<sup>34</sup>. A typical beam element is shown in Figure 8-3. It has length L and two nodes, one at each end of the beam. There are three degrees of freedom (DOFs) defined at each node i. These are: displacement in the x direction ( $u_i$ ), displacement in the y direction ( $v_i$ ), and rotation about the z axis ( $\theta_i$ ). There are a total of six DOFs per beam element, and thus the elemental stiffness matrices are of size (6 by 6).



**Figure 8-3:** Beam element

The case study structure is shown in Figure 8-4. It is modelled by beam elements of four different lengths. The structure has 13 elements and 11 nodes (the nodes are indicated by the small rectangles). With 3 DOFs per node, and 11 nodes, the structure has a total of 33 DOFs. Thus, the unconstrained structure stiffness matrix is of size (33 by 33). The node numbering indicated in Figure 8-4 is optimal for the minimization of the structure stiffness matrix bandwidth, which is 21 for this structure.

For a static analysis, the boundary conditions for the problem are imposed by constraining all the DOFs at the ground nodes (3,6,9) to be zero. This effectively removes the corresponding 9 rows and columns from the stiffness matrix, and thus the problem size is reduced to (24 by 24) (nine DOFs are removed from the 33 in the unconstrained problem). Static loads can be applied to the structure at any of the unconstrained nodes. The resulting linear static equation is



**Figure 8-4:** Case study structure model

$$\mathbf{Kd} = \mathbf{p}, \quad (8.9)$$

where  $\mathbf{K}$  is the structure stiffness matrix,  $\mathbf{d}$  is the vector of unknown displacements and rotations, and  $\mathbf{p}$  is the static load vector. Because this is a linear finite element problem formulation, any static analysis can be carried out independently of a dynamic analysis, and both results can be superimposed for a total analysis. Thus, since we have previously completed a case study involving a static finite element analysis (Chapter 5), we will only carry out a dynamic analysis for this case study.

### 8.4.1 Linear Dynamic Analysis

Dynamic analysis<sup>35</sup> of the structure in Figure 8-4 requires solution of the matrix equation  
 $M\ddot{d} + C\dot{d} + Kd = p(t)$ . (8.10)

A consistent mass matrix  $M$  is used in the analysis, and thus it has the same structure as the stiffness matrix  $K$ , as does the damping matrix  $C$ . Earlier, we reported that we would do the analysis without damping<sup>31</sup>. However, as the problem formulation progressed, the decision was made to include damping to yield more realistic results. The vector  $p(t)$  is a vector of time-varying loads, and  $\ddot{d}$ ,  $\dot{d}$  and  $d$  are the acceleration, velocity, and displacement vectors, respectively. We consider a linear analysis, i.e. the mass, damping, and stiffness matrices remain constant throughout the problem solution.

In our dynamic analysis, we will investigate the response of the structure to earthquake loadings. In such an analysis, the ground nodes cannot be constrained, as the earthquake is imparting forces and causing displacements, velocities, and accelerations at those nodes. Thus, boundary conditions are not applied in a conventional manner, and a different approach is used for the analysis, which is explained below.

An earthquake transfers energy from the movement of the earth to a structure, and the actual loading forces at the ground points depend on the structure and are not known a priori. Thus, an earthquake analysis is not usually performed by applying time-varying loads to the ground nodes of a structure. Instead, the time-histories of the displacements, velocities, and accelerations of the nodes due to an earthquake are prescribed from experimental and/or theoretical data. From this information, the movements of the other nodes in the structure are calculated.

We consider the general case where the ground nodes do not move uniformly, and set up the problem as follows. The nodal acceleration, velocity, displacement, and load vectors of

equation (8.10) are written in two partitions. The top partition of the vectors are those nodes where the accelerations, velocities, and displacements are unknown (not prescribed), and any static or time-varying loads are known. The bottom partition of the vectors consists of the nodes where the accelerations, velocities, and displacements are prescribed, and the nodal loads or forces are unknown. This partitioning of the matrix equation is illustrated below,

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{Bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \end{Bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{Bmatrix} \dot{d}_1 \\ \dot{d}_2 \end{Bmatrix} + \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix}, \quad (8.11)$$

where block row 1 is the partition of nodes with unknown accelerations velocities, displacements, and known loads, and block row 2 is the partition of nodes with prescribed accelerations velocities, displacements and unknown loads. The corresponding partitions of the mass, damping, and stiffness matrices are as indicated. To obtain the partitioning, rows of the original matrix equation are simply switched.

The first equation of (8.11) is now rearranged to yield the following matrix equation

$$M_{11}\ddot{d}_1 + C_{11}\dot{d}_1 + K_{11}d_1 = p_1 - M_{12}\ddot{d}_2 - C_{12}\dot{d}_2 - K_{12}d_2, \quad (8.12)$$

where the entire right-hand side is known. For our case study, equation (8.12) has matrices of size (24 by 24), and vectors of size (24 by 1). This matrix equation is solved for the acceleration, velocity, and displacement vectors  $\ddot{d}_1$ ,  $\dot{d}_1$ , and  $d_1$ . If the vector of forces,  $p_2$  at the ground nodes is desired, we may solve for it by using the bottom equation of (8.11), once  $\ddot{d}_1$ ,  $\dot{d}_1$ , and  $d_1$  are obtained.

We will solve equation (8.12) using Newmark's direct integration method. This solution method has been detailed earlier<sup>31</sup>, and may be found in various references<sup>35</sup>. The earthquake acceleration data will be generated by computer, and will contain frequency components appropriate for earthquake motion. The velocity and displacement data will be obtained by integrating the acceleration data. The earthquake data will simulate an earthquake of 5 to 20 seconds in duration, and appropriate time steps will be used in the Newmark algorithm. Most of the computational effort in such an algorithm is to compute the matrix-vector multiplies at each time step.

## 9. OPTICAL PROCESSING EXTENSIONS

In the third year of research we intend to pursue several extensions to our optical processing operations, including on-line arithmetic, polynomial evaluation and floating point operations. We now discuss each of these topics.

We have recently formulated the concept for a new optical processor which implements on-line arithmetic<sup>36</sup>. The performance of on-line arithmetic has been shown to be superior in terms of speed to conventional arithmetic in applications where computations are executed concurrently and where pipelining may be employed<sup>37</sup>. On-line algorithms for addition, subtraction, multiplication and division have been described in the literature<sup>38, 36</sup>. Conventional division is not suited to implementation in optical processors and only recursive division algorithms have been proposed in the literature.<sup>39</sup>. However, on-line division may be readily implemented on our proposed on-line optical processor which also performs on-line addition, subtraction and multiplication. This architecture represents a new approach to numeric computations in optical processing. We will continue during the third year to investigate our on-line arithmetic optical architecture as a fast, efficient processor of variable precision computations. We will explore in greater detail, design requirements of the processor to fully exploit the advantages of on-line arithmetic. The range of usefulness of our processor can be expanded by including on-line square-root algorithms and the evaluation of vector expressions.

Polynomial evaluation via on-line arithmetic has been proposed for digital computers<sup>40</sup>. We will investigate the design of the optical on-line architecture that will incorporate polynomial evaluation as well as the aforementioned on-line algorithms, thus providing a more general-purpose on-line optical processor.

A technique for implementing floating-point operations has been detailed for our optical processor<sup>41</sup>. This method handles the mantissa in the optical processor and the exponent in external hardware. During the third year of research we will investigate alternative methods of floating point implementation that may better exploit the optical nature of the processor.

## 10. SUMMARY AND FUTURE WORK

This report has described the progression of our research to date, and our remaining plans for the third year. The bulk of that effort will be to implement the new case studies on the existing processor and the AC-coupled version as described in Chapter 7. We also plan to developed a new digital simulation and new error models, as discussed in our recent Research Proposal. This will give us the ability to investigate multi-channel architectures and binary and multi-level data encoding, and to verify their performance.

## References

1. B.K. Taylor and D. Casasent, "Twos-Complement Data Processing for Improved Encoded Matrix-Vector Processors", *Applied Optics*, Vol. 25, 15 March 1986, pp. 961-965.
2. R.P. Bocker, S.R Clayton and K. Bromley, "Electrooptical Matrix Multiplication using the Twos Complement Arithmetic for Improved Accuracy", *Applied Optics*, Vol. 23, July 1983, pp. 2019-2021.
3. A.P. Goutzoulis, "Systolic Time-Integrating Acoustooptic Binary Processor", *Applied Optics*, Vol. 23, No. 22, November 1984, pp. 4095-4099.
4. Caroline Perlee and David Casasent, "Negative base encoding in optical linear algebra processors", *Applied Optics*, Vol. 25, January 15 1986, pp. 168-169.
5. D. Casasent and J. Jackson, "Space and Frequency-Multiplexed Optical Linear Algebra Processors: Fabrication and Initial Tests", *Applied Optics*, Vol. 25, 15 July 1986, pp. 2258-2263.
6. D. Casasent and B.K. Taylor, "Banded-Matrix High-Performance Algorithm and Architecture", *Applied Optics*, Vol. 24, May 1985, pp. 1476-1480.
7. "Special Issue on Optical Computing", *Proc. IEEE*, Vol. 72, No. 7, July 1984.
8. E. Swartzlander, "The Quasi-Serial Multiplier", *IEEE Trans. Comput.*, Vol. C-22, April 1973, pp. 317-321.
9. H.J. Whitehouse and J.M. Speiser, "Linear Signal Processing Architectures", *Aspects of Signal Processing - Part II*, G. Tacconi, ed., NATO Advanced Study Institute, Boston, MA, 1976, pp. 669-702.
10. D. Psaltis, D Casasent, D. Neft and M. Carlotto, "Accurate Numerical Computation by Optical Convolution", *Proc. SPIE*, Vol. 232, April 1980, pp. 151-156.
11. W.C. Collins, R.A. Athale and P.D. Stilwell, "Improved Accuracy for an Optical Iterative Processor", *Proc. SPIE*, Vol. 352, 1982, pp. 50-56.
12. D. Casasent, "Scene Analysis Research: Optical Pattern Recognition and Artificial Intelligence", *SPIE, Advanced Institute Series on Hybrid and Optical Computers*, March 1986, Leesburg, Virginia
13. D. Casasent and J. Jackson, "Optical Linear Algebra Processor: Laboratory System Performance for Optimal Control Applications", *Proc. SPIE*, Vol. 639, March-April 1986.
14. D. Casasent, J. Jackson and G. Vaerwyck, "Optical Array Processor: Laboratory Results", *Proc. SPIE*, Vol. 700, IOCC 1986, Jerusalem, Israel, July 6-11 1986.
15. D. Casasent and A.K. Ghosh, "Optical Linear Algebra Processors: Noise and Error Source Modeling", *Optics Letters*, Vol. 10, June 1985, pp. 252-254.
16. G.F. Carrier and C.E. Pearson, *Partial Differential Equations - Theory and Technique*, Academic Press, New York, 1976, p. 262
17. L.A. Hagemann and D.M. Young, *Applied Iterative Methods*, Academic Press, New York, 1981.

18. R. Vichnetvetsky, *Computer Methods for Differential Equations I*, Prentice-Hall, New York, 1981.
19. D. Casasent, A. Ghosh and C.P. Neuman, "A Quadratic Matrix Algorithm for Linear Algebra Processors", *J. Large-Scale Systems*, Vol. 91985, pp. 35-49.
20. H. Carslaw and J. Jaeger, *Conduction of Heat in Solids*, Clarendon Press, Second Edition, Oxford, 1959.
21. D. Casasent and J. Jackson, "Real-Time Optical Laboratory Linear Algebra Solution of Partial Differential Equations", *Proc. SPIE*, Vol. 698, August 1986.
22. D. Casasent and J. Jackson, "Laboratory Optical Linear Algebra Processor for Optimal Control", *Optics Communications*, Vol. 60, 15 October 1986, pp. 1-4.
23. J. Fisher, D. Casasent and C.P. Neuman, "Factorized Extended Kalman Filter for Optical Processing", *Applied Optics*, Vol. 25, 15 May 1986, pp. 1615-1621.
24. D. Casasent and C. Perlee, "Bipolar Biasing in High-Accuracy Optical Linear Algebra Processors", *Applied Optics*, Vol. 25, 1 April 1986, pp. 1033-1035.
25. B.K. Taylor and D. Casasent, "Error-Source Effects in a High-Accuracy Optical Finite-Element Processor", *Applied Optics*, Vol. 25, 15 March 1986, pp. 966-975.
26. AUGAT, "Flight Manual Designer's Guide", Manufacturers literature, 40 Perry Ave., Attleboro, MA , 1985, High speed Wire-Wrap designer's guide
27. TRW LSI Products Division, *VLSI Data Book*, TRW Inc. La Jolla, CA 92038, P.O. Box 2472, 1984, Pages 101-112
28. Honeywell, "HDAC34010 Triple 4-bit, high speed raster D/A converter", Datasheet, 1150 Cheyenne Mountain Blvd. Colorado Springs, CO 80906, 1985.
29. K. Kawano and O. Mitomi, "Coupling characteristics of laser diode to multimode fiber using separate lens method", *Applied Optics*, Vol. 25, No. 1, January 1986, pp. 136-141.
30. M. Sochacka, J. Sochacki and C. Gomez-Reino, "Paraxial region index profiles of GRIN rods", *Applied Optics*, Vol. 25, No. 1, January 1986, pp. 142-145.
31. D. P. Casasent, B. K. Taylor, C. Perlee - Carnegie Mellon University, "Parallel Optical Finite Element and Computational Fluid Dynamics Processors", Tech. report, Nasa Langley Research Center, March 1987.
32. Dr. Janet Peterson, University of Pittsburgh, PA.. Information obtained through oral discussion
33. R. E. Smith, A. Kidd, *Numerical Studies Of Incompressible Viscous Flow In A Driven Cavity*, NASA Langley Research Center, Washington, D.C., 1975, pp. 61-82, ch. 6.
34. J. S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw-Hill, New York, 1968.
35. K. J. Bathe and E. L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

36. M.J. Irwin, *An Arithmetic Unit For On-Line Computation*, PhD dissertation, University of Illinois at Urbana-Champaign, May 1977, Report No. UIUCDCS-R-77-873
37. M.D. Ercegovac, "On the performance of on-line arithmetic", *Proc. 1980 International Conference on Parallel Processing*, 1980, pp. 55-62.
38. K.S. Trivedi and M.D. Ercegovac, "On-line algorithms for division and multiplication", *IEEE Trans. on Computers*, Vol. C-26, July 1977, pp. 681-687.
39. H. John Caulfield, "Application of optical pipelines to root searching and to division", *Applied Optics*, Vol. 23, Feb. 1984, pp. 373-376.
40. M.D. Ercegovac, "A general hardware-oriented method for evaluation of functions and computations in a digital computer", *IEEE Trans. on Computers*, Vol. C-26, July 1977, pp. 667-680.
41. J. Fisher, *Extended Kalman Filter Algorithms for implementation on a High-Accuracy Optical Processor*, Carnegie-Mellon Univ., Pittsburgh, Pa., December, 1984, Masters Thesis.

# I. POST-DETECTION HARDWARE DESIGN

## I.1 Introduction

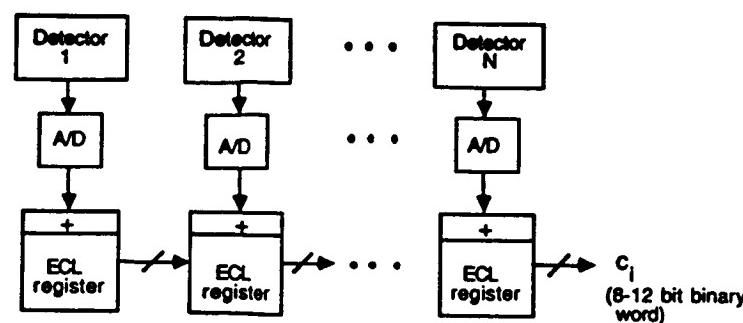
This Appendix discusses the hardware and/or software that could be used with the output detectors of the multichannel OLAP, to implement a higher level radix ( $\text{radix} > 2$ ) or a negative radix. We do not plan to use hardware but will use software/hardware for cost reasons. We do not detail here the software procedure as it is straight forward. Instead, we concentrate on the real-time hardware to demonstrate its feasibility. The extra hardware required (beyond what we presently plan to implement) would convert the mixed radix detector output into a binary word, so that this may be fed directly back into the controlling microprocessor. If this output is to be re-used as a multi-level input to the OLAP on the next cycle, as in various recursive algorithms, a D/A conversion will produce a higher-level encoding of the output binary word to the optical processor's input. In the existing laboratory OLAP, the input operands are encoded in conventional binary. In this case, simple shift/add hardware converts the mixed radix output back into binary. When a negative base encoding is used, or, when the radix is positive but not a power of 2, e.g.,  $\text{radix}=3$  or  $\text{radix}=5$ , then this simple shift procedure is not sufficient and we must resort to a slightly more complex shift/add procedure. The algorithms that define these procedures are detailed in Section I.2. Several modified OLAP detection systems are presented in Section I.3. These designs are presented as possible future modifications to the existing detection system but will not be implemented in the laboratory. The existing detection system will be used. Conclusions and a summary are presented in Section I.4.

## I.2 Basic OLAP output hardware

We begin by looking at the existing laboratory OLAP to show that the existing "back-end" is insufficient for handling negative radices and higher-level radices and that additional hardware will be needed in order to convert the mixed-negative-radix output into a *binary* word.

The same inadequacy holds for radices that are not a power of two, e.g. radix=3 or radix=5.

Figure I-1 shows a schematic of the current post-detection electronics.



**Figure I-1:** OLAP output configuration

Whether the OLAP is running in single- or multi-channel mode, positive or negative radix, the values from the  $N$  detectors of Fig. I-1 will be mixed radix, i.e. the values on the detectors may be greater than the radix magnitude. Every  $T_1$ , these mixed radix digits are A/D converted into a 6-bit binary word. Each 6-bit word is then added to the ECL register directly beneath each A/D (Fig. I-1). The contents of each ECL register are then shifted to the register at its right. The rightmost ECL register of Fig. I-1 shifts its contents (an 8 to 12 bit binary word labeled  $c_i$  in Fig. I-1) out as output. This  $c_i$  is a valid digit of the VIP result. Each  $c_i$  is output in bit-parallel format. The A/D-ECL combination hardware simulates a CCD shift register by adding successive A/D output data to prior accumulated and shifted output data in the ECL registers. In the next  $T_1$  cycle, new incident light is detected by the  $N$  detectors and the process is repeated. One valid binary word,  $c_i$ , is output from the rightmost register (Fig. I-1) every  $T_1$ . In other words, binary word  $c_0$  is output at time  $t=T_1$ , then  $c_1$  is output at  $t=2T_1$ ,  $c_2$  is output at  $t=3T_1$ , and so on. Each binary  $c_i$  is an 8 to 12 bit word and all the  $c_i$  combine to form the final output, which we will assume to be the scalar  $Z$  of the OLAP input data, as given by

$$Z = \sum_{i=0}^{N-1} c_i r^i , \quad (I.1)$$

where  $r$  is the encoding radix of the OLAP input data and  $N$  is the number of bits in the input operands and also the number of detectors, A/Ds and ECL registers. The  $c_i$  words are produced

sequentially, i.e. one every  $T_1$ , and the  $N$  least-significant digits of the  $Z$  are produced after time  $t=N \times T_1$ . The product of two  $N$ -bit operands is a  $2N$ -bit mixed-radix word. However, we retain only the  $N$  least-significant digits from the detector plane. If any of the  $N-1$  digits remaining in the ECL registers after time  $t=NxT_1$  are nonzero, this is treated as overflow and  $Z$  is set to its maximum possible value,  $r^{N-1}$ , where  $r$  is the encoding radix. This Appendix describes the hardware/software that will convert the  $N c_i$  words into a binary representation (one word) of the scalar  $Z$ . With  $Z$  in binary form, it can be used directly as input to the controlling microprocessor, or to a D/A convertor to generate a base  $r$  encoding of  $Z$  (for multilevel encoding of the OLAP).

There are two cases which we considered in the hardware design:

1. the encoding radix of the operands is positive or negative and is a power of 2, i.e.  $\text{radix}=\pm(2)^k$ , where  $k=1$  is conventional binary.
2. the encoding radix of the operands is positive or negative and is not a power of 2, e.g.,  $\text{radix}=\pm 3$ , or  $\text{radix}=\pm 5$ .

#### I.2.1 Case 1: radix is positive or negative and a power of 2

The first case is the simpler of the two cases to implement in hardware (in software, both cases are relatively simple to implement). In this case, the mixed-radix-to-binary conversion of the output  $c_i$  words is performed by simple shift/add/subtract combination hardware. The block diagram of Fig. I-2 illustrates the concept when the encoding radix is positive. Every  $T_1$ , the newest  $m$ -bit binary word  $c_i$  from the output of Fig. I-1 ( $m$  is 8 to 12 bits in the laboratory system) is loaded into the shift/add block in bit-parallel form. The subtract operation is not needed for positive radices and thus is not shown. We illustrate with the example shown to the right in Fig. I-2 where the OLAP encoding radix is binary, and  $N=4$ , i.e., the OLAP's input operands are 4-bit binary words, and the binary output words from the OLAP back-end of Fig. I-1 are  $c_0=0001_{(2)}$ ,  $c_1=0010_{(2)}$ ,  $c_2=0011_{(2)}$  and  $c_3=0100_{(2)}$ . (These were chosen arbitrarily for

this example). Each  $c_i$  is output from the ECL registers of Fig. I-1 in bit-parallel fashion and thus are bit-parallel loaded into the shift register of Fig. I-2. For this example  $Z = \sum_{i=0}^3 c_i 2^i = 1x2^0 + 2x2^1 + 3x2^2 + 4x2^3 = 49_{(10)} = 0110001_{(2)}$ . The last digit in this  $Z$ , the binary one, is the output from the shift/add block in Fig. I-2 after time  $t=NxT_1$ . As shown, the multiplication of  $c_i$  by  $2^i$  is analogous to shifting  $c_i$  one bit to the left, with respect to the previously generated  $c_{i-1}$ , and then adding. This shifting and adding process is performed by the shift/add block of Fig. I-2. The result from the shift/add block, after all  $c_i$  have been input, is the expected binary  $Z$ . The shift/add operation can be performed with a parallel-load shift register and binary adder.

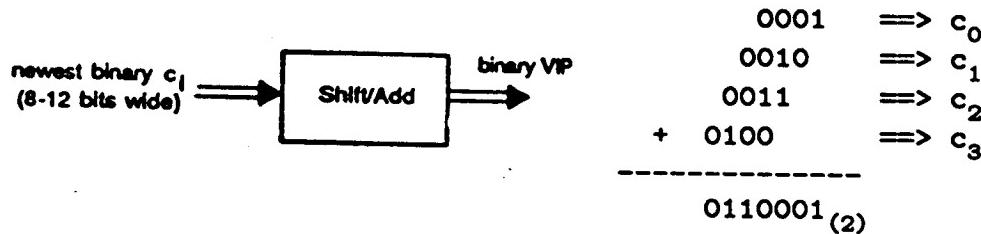


Figure I-2: Case 1: Forming binary  $Z$  from  $N$  binary words,  $c_i$

We now consider the case when the radix is positive and a power of 2, i.e.  $\text{radix} = 2^k$  (with  $k > 1$ ). Under these conditions, we still use the shift/add operation of Fig. I-2, but now each  $c_i$  is shifted by  $k$  bits with respect to the previous  $c_{i-1}$ , rather than a shift of one bit, as in the example of Fig. I-2. This represents only a trivial adjustment to the shift/add operation of Fig. I-2 and is readily incorporated into the system design of the shift/add hardware.

To summarize, we can readily generate the binary scalar  $Z$  from the  $N$  distinct binary words,  $c_i$ , when the encoding radix in the OLAP is conventional binary or a positive radix that is a power of 2,  $r = 2^k$ . We now discuss the case when the encoding radix is negative and a power of 2, i.e.  $r = -(2^k)$ .

To produce the binary  $Z$  from the  $N$  binary  $c_i$  words, when the OLAP encoding is a

negative base, the shift/add operation also requires a subtraction. To illustrate, we expand (I.1), for the case  $r = -(2^k)$ , into binary form and let the binary  $c_i$  be  $c_0=0001_{(2)}$ ,  $c_1=0010_{(2)}$ ,  $c_2=0011_{(2)}$  and  $c_3=0100_{(2)}$ , as in the previous example. Also, let the encoding radix of the OLAP be negabinary, i.e.  $r = -2$ . Hence, we express (I.1) as

$$Z = \sum_{i=0}^3 c_i (-2)^i. \quad (I.2)$$

As in the previous examples corresponding to a positive radix, (I.2) must be expressed in binary form because the shift/add/subtract hardware is binary-based. Thus, we expand (I.2) as

$$\begin{aligned} Z &= \sum_{i=0}^3 c_i (-2)^i = c_0 x(-2)^0 + c_1 x(-2)^1 + c_2 x(-2)^2 + c_3 x(-2)^3 \\ &= c_0 x(2^0) - c_1 x(2^1) + c_2 x(2^2) - c_3 x(2^3). \end{aligned} \quad (I.3)$$

The last line of (I.3) is analogous to the example at the right of Fig. I-2, except that the outputs  $c_i$  are alternately added and subtracted in (I.3). The amount of shift for each  $c_i$  in (I.3) is the same as in Fig. I-2. Therefore, when the encoding radix is positive, the  $c_i$  are shifted and added. When the encoding radix is negative, the  $c_i$  are shifted and alternately added and subtracted. Thus, the digital hardware realization of these shift/add algorithms has been designed to allow both positive and negative radix encoding, by employing a binary adder/subtractor or ALU in the circuit.

### I.2.2 Case 2: Radix is positive or negative and not a power of 2

The simple shift and add algorithm of Fig. I-2 is not applicable when the radix is not a power of 2, e.g., when  $r=\pm 3$  or  $r=\pm 5$ . In this case, in order to convert the  $N c_i$  words into the binary scalar  $Z$ , multiple shifts and adds must be performed on each  $c_i$ . By comparison, when  $r=2^k$ , only one shift (of  $k$  bits) is necessary for each  $c_i$  before it is added to the sum of the prior outputs. We illustrate with an example. Consider the case when the encoding radix is  $r=3$ . As in the example of Fig. I-2, we arbitrarily let the  $c_i$  words be  $c_0=0001_{(2)}$ ,  $c_1=0010_{(2)}$ ,  $c_2=0011_{(2)}$  and  $c_3=0100_{(2)}$ . Note that the  $c_i$  are binary, not base 3. No attempt has been made to alter the hardware design of Fig. I-1, which already exists in the laboratory set-up.

The scalar  $Z$ , as defined by (I.1), is

$$Z = \sum_{i=0}^3 c_i 3^i = c_0 x_1 + c_1 x_3 + c_2 x_9 + c_3 x_{27}. \quad (\text{I.4})$$

Recall that each  $c_i$  is an 8 to 12 bit binary word. As in the previous examples, we express (I.4) in binary base since the shift/add/subtract hardware is binary based, thus

$$Z = \sum_{i=0}^3 c_i 3^i = c_0 x_1 + c_1 x^{2^1} + c_2 x^{2^2} + c_3 x^{2^3}, \quad (\text{I.5})$$

$$= c_0 x^{2^0} + c_1 x(2^0+2^1) + c_2 x(2^0+2^3) + c_3 x(2^0+2^1+2^3+2^4). \quad (\text{I.6})$$

Equation (I.6) is obtained from (I.5) by replacing the base +2 representation for  $3^i$  in (I.5). Equations (I.5) and (I.6) are equivalent, except that (I.6) is directly implementable in digital shift/add hardware in a manner similar to the example of Fig. I-2. In other words, (I.6) is a sum of  $c_i x^{2^j}$  terms, and each  $c_i x^{2^j}$  can be produced by shifting  $c_i$  by  $j$  bits to the left (of its LSB). Let us now detail the shift and add realization of (I.6). In the first term, the multiplier  $2^0$  of  $c_0$ , indicates that no shift is performed on  $c_0$ . To this  $c_0 x^{2^0}$  term we add the term  $c_1 x(2^0+2^1)$ , which is produced by shifting  $c_1$  by 0 bits (i.e. unshifted) and adding that to  $c_1$  shifted by 1 bit. Proceeding similarly, appropriate shifts and adds on each of the  $c_i$  will produce the binary  $Z$  for  $R=3$ . Figure I-3 illustrates the corresponding shifts and adds necessary to produce the  $Z$  result in (I.6).

0 0 0 1	$c_0 (2^0)$
0 0 1 0	$c_1 (2^0+2^1)$
0 0 1 0	$c_2 (2^0+2^3)$
0 0 1 1	$c_3 (2^0+2^1+2^3+2^4)$
0 0 1 1	
0 1 0 0	
0 1 0 0	
0 1 0 0	
+ 0 1 0 0	
$1 0 0 0 1 1 1 0_{(2)} = Z_{(2)}$	

Figure I-3: Shift/Add procedure when  $R=3$

The method used in expanding the  $Z$  of (I.5) into the binary form of (I.6) can be generalized to any positive radix. Figure I-4 shows the binary expansion of a scalar  $Z$  for OLAP

encoding radices  $r=2,3,\dots,9$ , and for  $N=4$ -digit operands. The figure illustrates that the shift/add operations for radices that are powers of two are simpler than those when the radix is not a power of two. The number of shifts and adds on each  $c_i$  increases with increasing radix. Negative radices are not shown in Fig. I-4 but analogous conclusions may be drawn..

$$Z_{(2)} = \sum_{i=0}^3 c_i 2^i = c_0(2^0) + c_1(2^1) + c_2(2^2) + c_3(2^3)$$

$$Z_{(3)} = \sum_{i=0}^3 c_i 3^i = c_0(2^0) + c_1(2^0+2^1) + c_2(2^3+2^0) + c_3(2^4+2^3+2^1+2^0)$$

$$Z_{(4)} = \sum_{i=0}^3 c_i 4^i = c_0(2^0) + c_1(2^2) + c_2(2^4) + c_3(2^6)$$

$$Z_{(5)} = \sum_{i=0}^3 c_i 5^i = c_0(2^0) + c_1(2^2+2^0) + c_2(2^4+2^3+2^0) + c_3(2^6+2^3+2^2+2^1)$$

$$Z_{(6)} = \sum_{i=0}^3 c_i 6^i = c_0(2^0) + c_1(2^2+2^1) + c_2(2^5+2^2) + c_3(2^7+2^6+2^4+2^3)$$

$$Z_{(7)} = \sum_{i=0}^3 c_i 7^i = c_0(2^0) + c_1(2^2+2^1+2^0) + c_2(2^5+2^4+2^0) + c_3(2^8+2^6+2^4+2^2+2^1+2^0)$$

$$Z_{(8)} = \sum_{i=0}^3 c_i 8^i = c_0(2^0) + c_1(2^3) + c_2(2^6) + c_3(2^9)$$

$$Z_{(9)} = \sum_{i=0}^3 c_i 9^i = c_0(2^0) + c_1(2^3+2^1) + c_2(2^6+2^4+2^0) + c_3(2^9+2^7+2^6+2^4+2^3+2^0)$$

**Figure I-4:** Expressing  $Z$  of base  $r=2,3,\dots,9$  as base  $r=+2$ .

Consider the case when the encoding radix is a multilevel negative radix. As an example, we let  $r = -3$  and let the  $c_i$  words be the same as in the previous examples. Then  $Z$  is defined from (I.1) by,

$$\begin{aligned} Z &= \sum_{i=0}^3 c_i (-3)^i \\ &= c_0 x(-3)^0 + c_1 x(-3)^1 + c_2 x(-3)^2 + c_3 x(-3)^3 \\ &= c_0 x(1) + c_1 x(-3) + c_2 x(9) + c_3 x(-27) \\ &= c_0 x(2^0) - c_1 x(2^0+2^1) + c_2 x(2^0+2^3) - c_3 x(2^0+2^1+2^3+2^4). \end{aligned} \quad (I.7)$$

A comparison with (I.6) shows that the multiplications and additions required are analogous except for the alternating addition and subtraction in (I.7). So, both (I.6) and (I.7) can be

implemented with the same hardware employing a binary adder/subtractor or ALU to handle the addition and subtraction. Calculations have shown that the speed of this conversion circuit is too slow to operate efficiently with the OLAP when the radix is not a power of 2. Therefore, we plan to implement the shift/add/subtract hardware and employ only radices that are powers of 2, and either positive or negative. We expect no loss in processing capability from this decision. The shift/add/subtract unit is implemented with the OLAP detection system as shown in Fig. I-5.

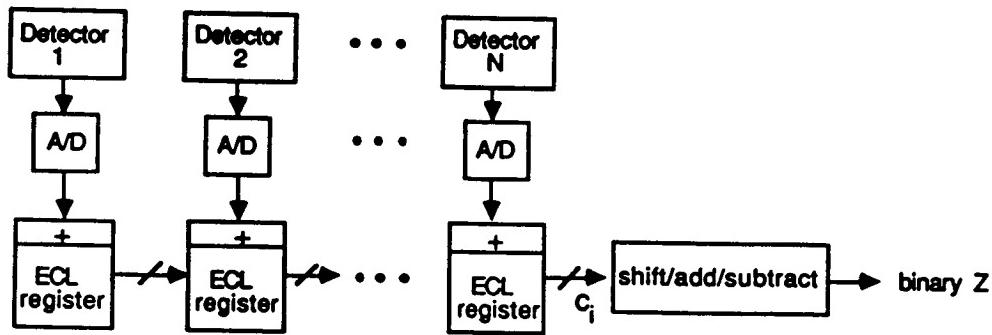


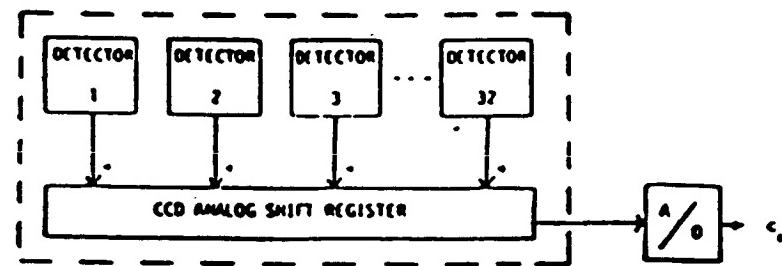
Figure I-5: OLAP detection system and conversion unit

### I.3 Future OLAP Detection Systems

The post-detection circuitry of Fig. I-1 has been constructed on the laboratory OLAP and will be used with all the tests conducted on the system. However, we propose several alternative designs to the post-detection circuitry to reduce its complexity while maintaining reasonable dynamic range requirements. Two of these alternatives are presented below.

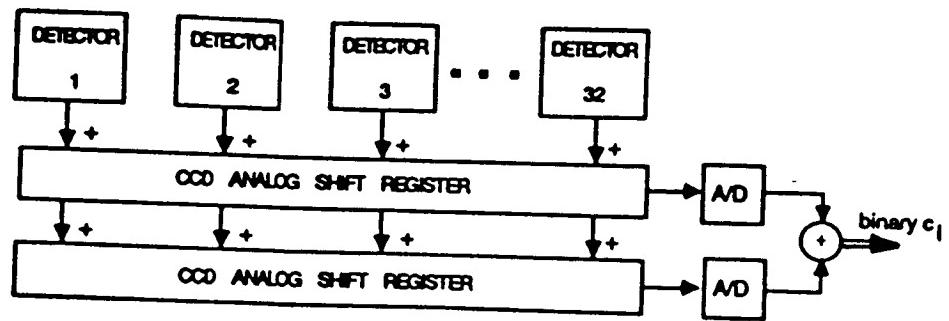
The post-detection circuitry of Fig. I-5 relies on the future availability of GaAs CCD analog shift registers. This design requires only one A/D. However, the dynamic range requirements are increased from that of Fig. I-1. The system of Fig. I-5 operates at the same throughput rate as the system of Fig. I-1. Every  $T_1$ , the rightmost, or current least significant,

mixed radix digit in the CCD shift register is shifted into the A/D. Its binary value,  $c_i$ , is output from the A/D and input to the shift/add/subtract hardware (not shown) as before. As in the present laboratory OLAP where the  $c_i$  are produced via the circuit in Fig. I-1, the shift/add/subtract hardware must perform all of the necessary shifts and adds corresponding to each  $c_i$  from Fig. I-6 in a time  $T_1$ , i.e. before the next  $c_{i+1}$  is generated by the A/D.



**Figure I-6:** Future back-end hardware

To reduce the dynamic range requirements of the CCD and the A/D of Fig. I-6 we further alter the post detection electronics by splitting the CCD and A/D into two levels, as shown in Fig. I-7. The operation of Figs. I-7 and I-6 are equivalent except that the top CCD in Fig. I-7 accumulates a charge up to some preselected threshold, then sends any excess accumulation to the bottom CCD in Fig. I-7. If the threshold is set to half the maximum charge allowed on the CCD of Fig. I-6, then each CCD and A/D of Fig. I-7 has half the dynamic range requirement of that in Fig. I-6. The A/Ds generate an output every  $T_1$  which are summed in a binary adder to form the binary  $c_i$  words. Thus, as in Figs. I-1 and I-6, a new binary  $c_i$  is produced every  $T_1$  (neglecting the time to sum the two A/D outputs).



**Figure I-7:** Future post-detection hardware with two-levels of CCDs

#### I.4 Conclusions and Summary

We have presented post-detection electronics that could be used with a multichannel OLAP employing a higher level radix or negative radix. The mixed radix detector values are converted into a binary word that is a  $Z$  of the input data and this binary word is converted into its equivalent value in the multilevel radix encoding of the original OLAP data, via a D/A conversion. We have shown that the electronics that generate the binary  $Z$  consist mainly of a shift register and adder/subtractor, and that this circuit works for all OLAP encoding radices, but that it is not practical, in terms of processing speed, when the encoding radix is not a power of 2. We thus restrict the OLAP to radices that are powers of 2, such as  $r = \pm 2, \pm 4$  or  $\pm 8$ . Calculations have demonstrated that under these conditions this hardware unit will not delay the throughput of the optical processor.

It is not our intent to build this shift/add/subtract unit. When high-level radices are employed, most of the operations will be performed in software. Software implementation is slower than in hardware but is simpler to initiate at this stage in the development of the OLAP. This discussion demonstrates, however, that a hardware implementation of the mixed-radix-to-binary conversion is simple and efficient.